Prince Mohammad Bin Fahd University

UNDERGRADUATE INFORMATION TECHNOLOGY AND COMPUTER ENGINEERING & SCIENCE PROGRAMS

PREFACE

The Undergraduate Information Technology and Computer Science Programs outline the four-year degree programs for students wishing to pursue undergraduate degree programs in Information Technology, Computer Science, or Computer Engineering within the College of Computer Eng. & Science at Prince Mohammad Bin Fahd University (PMU).

The integrated institutional structure is based on the North American model of education with English as the language of instruction. Distinguishing characteristics of the PMU, which set the university apart from existing institutions in the Kingdom of Saudi Arabia, represent a commitment to a set of competencies and learning outcomes that are integrated throughout the curriculum in a developmental manner.

The College of Computer Eng. & Science accepts successful male and female students from the PMU Preparation Year Program. The classroom experience for students in the College of Computer Eng. & Science is highly student-centered, interactive, and communicative. Syllabi include techniques for incorporating opportunities for students to develop communication, teamwork, and leadership skills as part of an overall strategy for achieving the PMU core competencies. Graduates from the College of Computer Eng. & Science will be self-directed, motivated, technically competent professionals with strong communication skills, capable of effective teamwork and leadership.

This volume presents the academic program structures within the College of Computer Eng. & Science and establishes the relationship between the degree program offerings within the college and the distinguishing PMU competencies.

_____

Issa Al Ansari, Ph.D.
University President
Prince Mohammad Bin Fahd University

TABLE OF CONTENTS

Item Page

I. Preface 2
II. PROGRAM DEFINITION 5
A. PURPOSE 5
B. VISION 5
C. MISSION 5
III. ADMINISTRATION AND FACULTY 6
A. COLLEGE ADMINISTRATION 6
B. DEPARTMENTAL ADMINISTRATION 6
C. FACULTY SELECTION 8
IV. STUDENT ENROLLMENT 10
A. STUDENT BENEFITS 10
B. PMU CORE COMPETENCIES 11
C. ADMISSIONS PROCESS AND REQUIREMENTS 12
D. PERFORMANCE EXPECTATIONS 13
V. THE EDUCATIONAL EXPERIENCE 14
A. TECHNOLOGY-INFUSED ENVIRONMENT 14
B. THE CLASSROOM EXPERIENCE 15
VI. DEPARTMENTS AND MAJORS 18
A. DEPARTMENT OF INFORMATION TECHNOLOGY 18
B. DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING 23
CS - CURRENT DEGREE PLAN 25
CE - CURRENT DEGREE PLAN 29
VII. LABORATORY REQUIREMENTS 31
A. GENERAL PURPOSE / COMPUTER LABORATORIES 31
B. SPECIAL PURPOSE LABORATORIES 33
VIII. COURSE SYLLABI 35
A. COURSE NUMBERING SYSTEM 35
B. GENERAL INFORMATION TECHNOLOGY COURSES 36
GEIT 1411: Computer Science I 37
GEIT 1412: Computer Science II 42
GEIT 2421 Data Structures 47
GEIT 2291: Professional Ethics 53
GEIT 2331: Mathematical Reasoning & Algorithmic Thinking 58
GEIT 3341: Database I 62
GEIT 3331: Computer Organization 66
GEIT 3351: Principles of Software Engineering 72
GEIT 4361: Practical Training 78
ASSE 4311: Learning Assessment III 81
C. INFORMATION TECHNOLOGY COURSES 87
ITAP 4364: Performance Analysis of Computer Networks (Never offered) 88
ITAP 1311- Introduction to IT 88
ITAP 2312: Web Programming 91
ITAP 2431: Network Management 95
ITAP 3431: Network Security 99
ITAP 3313: User Interface Development 104
ITAP 3471: Web Server Administration 109
ITAP 3411: Systems Programming 115
ITAP 3382: Business Intelligence 121
ITAP 3383: Enterprise Resource Planning Systems 127
ITAP 4371: E-Commerce 133
ITAP 4316: Introduction to Software Project Management 137

ITAP 3371: Database II 144
ITAP 4313: Advanced Visual Programming 149
ITAP 3362: Mobile Application Design & Development 153
ITAP 4315: Cloud Computing 156
ITAP 4361: Operating Systems 161
ITAP 4390: Information Security and Assurance 166
D. COMPUTER SCIENCE COURSES 170
COSC 2312: Web Programming 171
COSC 3332: Discrete Structures and Combinatorial Analysis 175
COSC 3361 : Computer Networks 179
COEN 3361: Computer Networks 179
COSC 3351: Algorithms 182
COSC 3411: Systems Programming 187
COSC 4461: Programming Languages 198
COSC 4362: Artificial Intelligence 204
COSC 4363: Theory of Computation 208
COSC 4364: Compilers 212
COSC4373: Computer Vision 217
COSC 3354: Introduction to Cryptography 220
COSC 4371: Computer Graphics 224
COSC 4374: Computer Vision 230
COSC 4311: Parallel Computing 233
ITAP 4371: E-commerce 243
ITAP 4316: Introduction to Software Project Management 247
ITAP 3371: Database II 254
E. COMPUTER ENGINEERING COURSES 259
COEN 2411: Circuits 260
COEN 3323: Digital and Logic Design 264
COEN 3421: Electronics 268
COEN 3361: Computer Networks 272
COEN 4331: Microprocessors 275
COEN 4322: Digital Signal Processing 279
COEN 4361: Operating Systems 284
COEN 4413: Embedded Systems 288
COEN 4331: Microprocessors 292

II. PROGRAM DEFINITION

A. PURPOSE

The College of Computer Eng. & Science will provide the structure and organization for male and female students to successfully pursue degree programs in Information Technology, Computer Science, and Computer Engineering at the undergraduate level.

B. VISION

The College of Computer Eng. & Science will provide a unique and distinguished academic unit that participates in:

- Preparing future Information Technology and Computer Science and Engineering professionals and leaders who can support the emergence of Saudi Arabia as a global IT resource.
- Enriching and developing Information Technology intellectual resources.
- Exploring innovative instructional methodologies and technologies to provide the highest quality effective preparation of information technology professionals.
- Establishing communication and the exchange of ideas between the academic and business communities.

C. MISSION

The College of Computer Eng. & Science will achieve the following objectives:

- Contribute to advancement of human intelligence and to the promulgation and development of knowledge and understanding in the Information Technology domain.
- Prepare professionals in Information Technology and Computer Science and Engineering, through the utilization of innovative educational processes, in a modern technological environment.
- Transform the graduate to play a pioneering and leading role in the community, enabling him or her to take responsibilities and contribute to solving problems through innovative thinking, collective work, reflection, and self-development.
- Link academic programs and specializations with actual requirements of the surrounding work environment. This will be achieved by maintaining effective partnerships between the university and local business and industry.
- Guide research activities to create solutions for persistent problems in surrounding communities through applied research and technical consultation. The importance of performing basic scientific research for enriching human intelligence should not be neglected.
- Provide community service through continuous training and education.

III. ADMINISTRATION AND FACULTY

A. COLLEGE ADMINISTRATION

The College of Computer Eng. & Science will fall under the authority of the Vice Rector of Academic Affairs and will be administered by the Dean of the College of Computer Eng. & Science.

The College of Computer Eng. & Science will be responsible for the organization and administration of three-degree programs:

- Bachelor of Science in Information Technology
- Bachelor of Science in Computer Science
- Bachelor of Science in Computer Engineering

(Detailed discussion of the duties of the Vice Rector of Academic Affairs, the Dean of the College of Computer Eng. & Science and the Chairs and Associate Chairs of the college's departments is provided in the report PMU Organization.)

B. DEPARTMENTAL ADMINISTRATION

Responsibility and authority for the daily operation of the college's three degree programs will lie in its two departments: the Department of Information Technology and the Department of Computer Science and Engineering.

- The Department of Information Technology will be responsible for the operation, administration and management of the degree program, including degree-specific and elective requirements, for the Bachelor of Science in Information Technology. It also will administer the general IT (GEIT) courses that are required of all students in the college.
- The Department of Computer Science and Engineering will be responsible for the operation, administration and management of the degree programs, including degree-specific and elective requirements, for the Bachelor of Science in Computer Science and the Bachelor of Science in Computer Engineering.

In each department, a Chair will oversee instruction of male students and an Associate Chair will oversee instruction of female students.

1. The Two-Department Structure

Though the college will offer three major fields of study, this report recommends the establishment of two, rather than three, departments for a number of reasons. Two departments will:

- Lower the cost of administration. This organization will require only two Chairs and two Associate Chairs for the departments.
- Encourage cooperation and coordination. In some institutions, programs in computer science and computer engineering find themselves competing for resources. One department overseeing two programs will encourage cooperation.

2. Departmental Responsibilities

The departments within the College of Computer Eng. & Science will set the tone for the entire college, including the relationships among faculty, students, and potential employing organizations. Smoothly run operations, therefore, will be essential to the success or the program. Each department will be responsible for:

- Appropriate academic advising for students: The department will strive to provide academic advising to students on an individual basis so as to determine the most appropriate course of study.
- Tutoring and remediation: In cooperation with the PMU Learning Resources Center, the faculty of the college will create tutoring and supplementary instructional programs to assist students who need extra assistance with academic programs or study skills. (A detailed discussion of such offerings and the organization that will provide them is provided in the report Learning Resources Center.)
- Maintenance and development of the curriculum. The department will manage continuous curriculum review and improvement. This function will be primarily the responsibility of the professorial faculty.
- Provision of course materials to students. Each student should be provided with all course materials by the program administration. These materials will include: textbooks, cases, articles, and in general any readings that the students are expected to prepare. Providing these materials will ensure that all students will receive the same material, will protect the copyrights of the material, and will be an added benefit to the students.
- Maintaining the class calendar. The calendar for each class of entering students will be published and followed from the first day of each academic semester. This calendar will show class meeting dates. It also will let students know in advance the dates for which they must prepare materials.
- Evaluation of faculty. The department will be responsible for the implementation of PMU policies and procedures for the evaluation of faculty. Each department will be responsible for providing appropriate data and information to the College of Computer Eng. & Science and to the university as required.

C. FACULTY SELECTION

The quality of faculty will be a critical component of the quality and success of each degree program within the College of Computer Eng. & Science. Faculty will be academically well prepared and will be proven effective teachers. Faculty will be able to demonstrate a history and currency in providing quality education that aligns well with the PMU core competencies and with the PMU educational philosophy and methods that provide a student-centered and positive environment. The criteria for faculty will be:

1. Degrees and Teaching Experience

   Faculty appointed to Professorial ranks will hold a Ph.D. in a relevant discipline. Faculty appointed to Instructor ranks will hold at least a Masters degree in a relevant field and should have experience of working in a professional environment. Upper division (3000 and 4000 level courses) will normally only be taught by faculty in Professorial ranks.

   The college will strive to maximize the proportion of faculty with terminal degrees. The college will also ensure those faculties have significant prior teaching experience at the university level. This requirement can be established through statements of teaching philosophy and through demonstration of teaching techniques during an interview.

   Preference will be given to faculty who possess prior experience in teaching in cooperative and collaborative learning environments.

2. English Language Skills

   All faculties will be proficient in the English language. Preference will be given to faculty who are either native English speakers or have achieved native-level proficiency as demonstrated by a band score of 8.0 or higher on the IELTS, with minimum component test scores of at least 7.5 (or equivalent score on a comparable exam).

3. Alignment to PMU Values

The PMU has established defining institutional characteristics that have a significant impact on the nature of the university and its degree programs. These characteristics also will impact the nature of faculty who will teach within the college. Those characteristics include:

- A student-centered approach to education and instruction including utilization, engagement, feedback and repetition (as identified in the report Undergraduate Core Curriculum Design).
- A willingness to undertake professional development activities necessary to learn how to implement a student-centered, communicative classroom environment. Such activities will be supported by the PMU Teaching Development Center.
- A personal and professional commitment of lifelong learning. Faculty will promote lifelong learning attitudes and concepts, not only through their teaching, but also by modeling such attitudes by their personal and professional continuing education activities.
- Sensitivity to Arab cultural and Islamic religious practices and expectations.

4. Student/Faculty Ratios

In order to enhance opportunities for class participation and individual attention, the student/faculty ratio in classes and labs of College of Computer Eng. & Science will be kept as low as possible.

The College of Computer Eng. & Science will maintain a maximum student/faculty ratio of approximately 24/1 for lecture and laboratory classes. The college will establish a general class size maximum of 30 students for any single non-laboratory class. Classes that have laboratory components should be restricted to no more than 24 students in order to ensure that appropriate facilities are available to those classes. The college will further work to ensure that the largest classes are distributed across the faculty to minimize inequalities in workload.

The two introductory courses GEIT 1411: Computer Science I and GEIT 1412: Computer Science II may be taught via a combination of large lectures and smaller sections at the discretion of the Departments of Information Technology and Computer Science and Engineering. Calculus courses will be taught via a combination of lecture classes and smaller recitation sections.

IV. STUDENT ENROLLMENT

A. STUDENT BENEFITS

In order to meet the established and growing needs for skilled computer professionals, the PMU College of Computer Eng. & Science will seek to achieve a number of learning objectives for its graduates.

Students graduating from the College of Computer Eng. & Science will be well prepared professionally, capable of meeting or exceeding the demands of professional employment in the region, and committed to continued professional development and improvement.

Students will graduate with the skills and experiences necessary to be technically competent in their field; capable of effective communication, cooperation and collaboration with professional peers; and able to provide leadership where appropriate.

The degree programs within the College of Computer Eng. & Science will:

- Create an understanding of the concepts and practices in the information technology professional fields and provide practical skills in support of those practices.
- Foster teamwork and leadership in the professional environment.
- Enhance the professional communication skills of the participants.

Students graduating from PMU with a degree from the College of Computer Eng. & Science will have the appropriate background, experiences and qualifications to immediately enter the professional workplace in Information Technology, or Computer Science and Engineering. There is significant and well-established current need for professionals in information technology, and computer science and engineering. The Kingdom of Saudi Arabia, through its Human Resources Development Fund, also has targeted IT as a significant part of its development strategy. These trends can have the potential consequence of creating greater demand for graduates with IT and CSE skills and qualifications

B. PMU CORE COMPETENCIES

As they work toward achieving the educational benefits outlined above for students, each of the degree programs within the College of Computer Eng. & Science will maintain values consistent with the undergraduate goals of the university. The development of six distinctive competencies (discussed in detail in the report Undergraduate Core Curriculum) is considered to be of value to all effective professionals, whether they are advancing their education at the graduate or undergraduate levels. The six PMU defining competencies are:

- Communication
- Technological competence
- Critical thinking and problem solving
- Professional competence.
- Teamwork
- Leadership

1. Building Competencies in the Classroom

The College of Computer Eng. & Science will provide an environment in which these core competencies will be recognized for their importance and centrality to the degree programs as well as actively pursued within each degree program. The course syllabi within each of the degree programs (see Section VIII, Course Syllabi, of this report) include specifications of the learning environment and activities and assignments designed specifically to:

- Foster improved communication through classroom and other presentations and through discussion and critique of those presentations.
- Enhance analysis, synthesis and other critical thinking and problem solving components through programming and other projects throughout the curriculum. These competencies will be addressed specifically through the three-course capstone sequence GEIT 3351: Principles of Software Engineering and ASSE 4311: Learning Outcome Assessment III.
- Provide opportunities for teamwork and leadership skills to be practiced through group projects within much of the curriculum and particularly within the capstone sequence.
- Require reflective self-assessment by students through the use of journaling techniques in much of the curriculum.

2. Building Competencies in the Community

In order to simultaneously build student competencies and meet the PMU goal of serving the surrounding community, the College of Computer Eng. & Science will offer a mandatory course internship in which students may receive academic credit for working in local business and industry.

Titled Practical Training, the course will require students to apply PMU competencies including professional competence, critical thinking, communication, leadership and teamwork. Faculty will supervise the internship and provide feedback on the student's performance and competencies.

A student who wishes to take the Practical Training but who is unable to arrange an internship in business or industry may be allowed to take a directed study course that will provide practical experience under the supervision of a faculty member.

Additionally, it is recommended that the PMU establish a Community Technology Resource Centre, which will provide students with work opportunities and will provide members of the community with sources of professional advice for small technology projects.

The centre will be jointly operated by the PMU College of Computer Eng. & Science and the PMU Centre for Research Development and Continuing Education. It will be run as a non-profit business, though fees will be charged to cover overhead, pay student wages, and provide additional income for faculty supervisors.

C. ADMISSIONS PROCESS AND REQUIREMENTS

The character and quality of students entering the College of Computer Eng. & Science will define the quality of the degree programs within the college as well as the quality of the graduates entering Information Technology professional environments after successfully completing the PMU program.

Admissions to the College of Computer Eng. & Science will be open to students who have successfully completed the PMU Preparation Year Program or who have met the university criteria for bypassing the program.

The degree programs in the College of Computer Eng. & Science will be designed to accept both male and female students.

1. Required Courses in the Preparation Year Program
   The PMU Preparation Year Program (as described in the report Preparation Program Design) concentrates on English language, mathematics, and study skills and learning strategies. English language, study skills, and the first semester math course, PRPM 0011: Introductory Algebra, are required of all students. However, during the second semester of mathematics, students have a choice of two tracks, depending on their desired major at the university.

   Students seeking entrance to the College of Computer Eng. & Science will be required to take PRPM 0022: Pre-Calculus, during the second semester of the Preparation Year Program.

2. Application for Admission
   Upon completion (or waiver) of the Preparation Year Program, students make application to the college in which they wish to study. This application will include:
   Preparation Year Program Certificate of Completion
   PMU Placement Test results
   Interview with the college
   Essay on a topic assigned by the college

   (A detailed discussion of admissions requirements and procedures is contained in the report PMU Admissions Plan.)

D. PERFORMANCE EXPECTATIONS

The College of Computer Eng. & Science will provide for minimum standards of academic performance from its students. Using a 4.0 scale for course grades, the College of Computer Eng. & Science will require that students maintain minimum grade point averages (GPA) for various categories of courses consisting of:

- 2.0 GPA in courses from the PMU Core Curriculum
- 2.0 GPA in degree-specific courses (courses from the Core Curriculum that IT students must complete beyond the minimum requirement)
- 2.25 GPA in courses required by the college (GEIT prefix)
- 2.5 GPA in courses within the academic discipline

A student who receives a D or F in any course will be required to repeat the course (in the case of an elective, another elective may be selected) and to achieve the required grade point average for that category of course. These students will be required to participate in tutoring and remediation programs offered by the college faculty and the PMU Learning Resources Center. (See Section III. B. 2, Departmental Responsibilities, Tutoring and Remediation, above.)

Students may repeat a course one time, with additional repeats allowed at the discretion of the faculty. However, no more than 10 repeated courses will be allowed over the student's career at the PMU. After the first repeat, prior grades will count toward the student's GPA. For example: A student who receives a D followed by an A will have the D erased and replaced with the A on the transcript. A student who receives an F followed by a D followed by an A will have the F erased, and both the D and the A will be averaged into the GPA.

In order to graduate, all students at the PMU will be required to maintain an overall GPA of 2.0.

V. THE EDUCATIONAL EXPERIENCE

A. TECHNOLOGY-INFUSED ENVIRONMENT

Information technology is central and critical to all degree programs in Information Technology. This will be especially true at the PMU where technology competencies are a hallmark of the successful student, and a technology-infused environment is a distinguishing characteristic of the university.

In the College of Computer Eng. & Science, the quality of access to technology will be a primary determinant in the quality of the educational experience of the student and the eventual professional competence of its graduates. The college will base its assumptions concerning learning and the learning experience on the universal availability of technology resources at all points on the university campus and from outside the campus through Internet-based resources.

1. Technology and the Classroom

   Access to technology within the classroom will be a necessary component of the degree programs within the College of Computer Eng. & Science. Faculty and students involved in classroom presentations will have access to modern presentation technology connected to university computing and library resources as well as to the Internet. (Facilities recommended for "smart" classrooms are discussed in the report PMU Infrastructure Specifications.)

2. Student Computing Requirements

   Like all other students at the PMU, students within the College of Computer Eng. & Science will be required to have personal laptop computers. They will have access to the university-wide technology-infused environment including wireless Internet access. However, students in the college will have specific computing requirements that extend beyond the standard Microsoft Office applications of a typical laptop. They will require access to compilers, design tools, and specialized computing environments.

Many of these specific computing requirements must be available through the university's technology infrastructure to students' laptop computers. Specialized computing facilities including access to distributed and parallel computing facilities and isolated network resources will be provided in general access and specialized laboratory facilities. (For a more detailed discussion of the college's specialized laboratory requirements see Section VII., Laboratory Requirements, in this report.)

Technologies such as interactive television, video conferencing and Blackboard or WebCT will be central to maintaining effective communication between faculty and students and among students. The College of Computer Eng. & Science will make extensive use of these resources to provide equal opportunities in learning within the context of gender separation. The college also will provide for student-oriented discussion through Instant Messaging and online discussion groups. Technology will enable students to directly submit materials, assignments and examinations, and to receive efficient communication of grades and faculty instructions.

The majority of major textbook publishers today provide electronic supplements to their books. Most of the textbooks recommended for the degree programs in the College of Computer Eng. & Science include such supplements, which the instructor may choose to use as appropriate.

B. THE CLASSROOM EXPERIENCE

The College of Computer Eng. & Science will make full use of specific classroom characteristics that reflect the defining characteristics of the university. These characteristics will include:

- A technology-infused classroom experience.
- A practical and hands-on orientation to the curriculum, including many laboratory-based classes.
- A curriculum that values teamwork through the use of group assignments and laboratory-based projects.
- A curriculum that values student communication through classroom-based presentations by students and ensuing class discussions.
- A curriculum that values formative self-assessment through the use of journaling and portfolio assessment.

These characteristics are implemented through specific formative and summative assessment requirements as described in individual syllabi. (Syllabi for courses offered by the College of Computer Eng. & Science appear in Section VIII., Course Syllabi, of this report.)

1. The Degree Programs

Each degree program will consist of a minimum of 128 semester credit hours in conformity to standards typical of American universities as specified in the document Defining Characteristics and Critical Path.

Each of the degree programs offered within the College of Computer Eng. & Science will consist of five components:

- General Education Requirements. These requirements for the University Core Curriculum and College Core Curriculum include 60 credit hours of courses in the PMU core competencies, communication, Arabic Language and Islamic Studies, physical education, mathematics, laboratory science, and social and behavioral sciences. (A detailed discussion of these requirements appears in the report Undergraduate Core Curriculum Design.)
- Degree-Specific Requirements. Each degree program will have requirements for additional courses from the College Core Curriculum in support of the degree program. Those courses will primarily be drawn from mathematics and laboratory science courses taught by the Core Curriculum faculty and business courses taught by the College of Business Administration. The degree-specific requirements are unique to each degree program.
- College of Computer Engineering & Science Requirements. These requirements will consist of nine courses totaling 29 credit hours that are common to all degree programs within the College of Computer Eng. & Science. They represent a base of knowledge that is presumed for all IT and computing professionals. The courses within the College of Computer Eng. & Science that meet these requirements are designated with the prefix GEIT. The GEIT courses include the following:
  - GEIT 1411: Computer Science I
  - GEIT 1412: Computer Science II
  - GEIT 2421: Data Structures
  - GEIT 2291: Professional Ethics
  - GEIT 2331: Math. Reasoning &Alg. Thinking
  - GEIT 3311: Computer Organization
  - GEIT 3341: Database I
  - GEIT 3351: Principles of Software Engineering
  - GEIT 4361: Internship

These common courses will administered by the Department of Information Technology.

- Degree Program Requirements. Each degree program will have unique requirements that differentiate the program from others within the college.
- Electives. Each degree program will identify the available electives and any constraints that will apply to the elective selection.

2. The College of Computer Eng. & Science Capstone Series

One of the critical components in the degree structure with the College of Computer Eng. & Science will be the capstone series, a sequence of three courses beginning in the junior year that will integrate conceptual materials and practical experience in the development of professional grade application development.

Building on the Capstone Series required by the PMU Core Curriculum (which begins in the sophomore year with ASSE 2111: Learning Outcome Assessment I and continues in the junior year with ASSE 3211: Learning Outcome Assessment II), the series will comprise three courses, GEIT 3351: Principles of Software Engineering and ASSE 4311: Learning Assessment III. Each course in the series will center on a different facet of software engineering.

- GEIT 3351: Principles of Software Engineering will examine the theory and practice of software development and maintenance with the focus being on the full software development life cycle, including coverage of tools, techniques, principles, and guidelines for software requirements, specification, design and implementation.
- ASSE 4311: Learning Assessment III will concentrate on the implementation, testing, debugging and maintenance of a designed software engineering solution.

The series of courses will lead students through conceptual understanding of how professional software development is managed and the tools and techniques used to do so.

VI. DEPARTMENTS AND MAJORS

A. DEPARTMENT OF INFORMATION TECHNOLOGY

The Department of Information Technology will be administered as a division of the College of Computer Eng. & Science. As such, it will fall under the senior leadership of the Vice Rector of Academic Affairs and the Dean of the College of Computer Eng. & Science.

Responsibility for the operation, administration and management of the Department of Information Technology will rest in the Chair of the Department of Information Technology. The chair will oversee the department as a whole, with primary responsibility for instruction of male students. He will delegate responsibilities as appropriate to the Associate Chair, who will oversee instruction for female students.

The Department of Information Technology is charged with the responsibility of maintaining, administering and managing the College of Computer Eng. & Science degree requirements; those requirements that are common to all degree programs within the college; and the degree requirements of the Bachelor of Science in Information Technology.

1. Content of the Information Technology Degree Program

   The Bachelor of Science in Information Technology is comprised of five components:

   - The PMU Core Curriculum. This core curriculum consists of 60 hours of coursework as defined in the report Undergraduate Core Curriculum Design.
   - Degree-Specific Requirements. These requirements represent support courses in mathematics, laboratory science and business. These requirements both specify and extend Core Curriculum requirements. The degree-specific requirements add 9 credit hours to the degree program.

     The University Core Curriculum requires six semester hours of mathematics. The Information Technology degree program extends that requirement to nine semester credit hours of mathematics and specifies that the courses will be:

     – MATH 1311 Finite Math
     – MATH 1312 Calculus for Business
     – MATH 1313 Statistical Methods

     University Core Curriculum requires eight semester hours of Natural and Physical Science. The Information Technology degree program extends that requirement to 8 semester credit hours and specifies that the courses will be:

     – Natural Science elective I
     – Natural Science elective II

- The College of Computer Eng. & Science Requirements. These requirements consist of 29 hours of coursework contained in the nine college courses designated with the GEIT prefix. This coursework represents conceptual and skill-based knowledge that is common to all degree programs within the College of Computer Eng. & Science. (See Section V.B.1., Degree Programs, of this report.)
- The Degree Program Requirements. These requirements consist of 27 hours of coursework as follows:
    - ITAP 1311: Introduction to IT
    - ITAP 2312: Web Programming
    - ITAP 2431: Network Management
    - ITAP 3431: Network Security
    - ITAP 3313: User Interface Development
    - ITAP 3471: Web Server Administration
    - ITAP 3411: Systems Programming
    - ITAP 3382: Business Intelligence
    - ITAP 3383: Enterprise Resource Planning Systems
    - ITAP 4371: E-Commerce

- Electives: The Information Technology degree program will require three semester credit hours from the College of Business:
    - Any 400 level course from the Management Information Science department, or
    - Any 400 level course from the College of Business

    The Information Technology Degree Program will require nine semester credit hours of elective to be taken from an approved list of IT electives.

A proposed model eight-semester course sequence follows:

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

FRESHMAN YEAR							
FIRST SEMESTER				SECOND SEMESTER			
Course Number	Course Title	Credit Hours	Pre-requisite	Course Number	Course Title	Credit Hours	Pre-requisite
ALIS 1211	Introduction to Islamic culture	2		ALIS 1212	The Social System in Islam	2	
COMM 1311	Written Communication	3		COMM 1312	Writing and Research	3	COMM 1311
UNIV 1211	Prof. Development and Competencies	2		UNIV 1212	Critical Thinking and Problem Solving	2	
MATH 1311	Finite Math	3	PRPM 0012	MATH 1312	Calculus for Business	3	MATH 1311
ITAP 1311	Intro to IT	3		GEIT 1412	Computer Science II	4	GEIT 1411
GEIT 1411	Computer Science I	4		PHED 1112	Healthy Behaviors and Management	1	
PHED 1111	Active Living Lifestyle	1					

SOPHOMORE YEAR							
FIRST SEMESTER				SECOND SEMESTER			
Course Number	Course Title	Credit Hours	Pre-requisite	Course Number	Course Title	Credit Hours	Pre-requisite
ALIS 2211	Linguistic Communication Skills	2		ALIS 2212	The Biography of Prophet Mohammad	2	
UNIV 1213	Leadership and Teamwork	2		COMM 2312	Technical and Professional Communications	3	COMM 2311
ASSE 2111	Learning Outcome Assessment I	1	Sophomore Level	ITAP 2312	Web Programming	3	GEIT 1411
COMM 2311	Oral Communication	3	COMM 1312	GEIT 2291	Professional Ethics	2	
GEIT 2421	Data Structures	4	GEIT 1412	GEIT 2331	Mathematical Reasoning & Algorithmic Thinking	3	GEIT 1412
MATH 1313	Statistical Methods	3	PRPM 0012	ITAP 2431	Network Management	4	GEIT 1412
	Social Science Elective I*	3					
Total Credit Hours		18		Total Credit Hours		17	

JUNIOR YEAR							
FIRST SEMESTER				**SECOND SEMESTER**			
Course Number	Course Title	Credit Hours	Pre-requisite	Course Number	Course Title	Credit Hours	Pre-requisite
ASSE 3211	Learning Outcome Assessment II	2	*ASSE 2111*	**ITAP 3471**	Web Server Administration	4	*ITAP 2312 ITAP 3411*
GEIT 3341	Database I	3	*GEIT 1412*	**ITAP 3411**	Systems Programming	4	*GEIT 3331*
ITAP 3431	Network Security	4	*ITAP 2431*	**ITAP 3383**	Enterprise Resource Planning Systems	3	*GEIT 3341 GEIT 1412*
	IT Elective I***	3		**GEIT 3351**	Principles of Software Engineering	3	*GEIT 1412*
GEIT 3331	Computer Organization	3	*GEIT 1412*	**ITAP 3382**	Business Intelligence	3	*GEIT 3341*
ITAP 3313	User Interface Development	3	*GEIT 1412 ITAP 2312*				
Total Credit Hours		**18**		**Total Credit Hours**		**17**	

SUMMER OF JUNIOR YEAR				
Course Number	Course Title	Credit Hours		Pre-requisite
GEIT 4361	Internship	3	8 weeks (320 hours) full time	End of Junior Year (summer before graduation) and department approval

SENIOR YEAR							
FIRST SEMESTER				SECOND SEMESTER			
Course Number	Course Title	Credit Hours	Pre-requisite	Course Number	Course Title	Credit Hours	Pre-requisite
	Natural Science Elective I**	4			IT Elective III***	3	
ITAP 4371	e-Commerce	3	ITAP 2312		Social Science Elective II*	3	
	IT Elective II***	3			MIS/Business Elective ****	3	
ASSE 4311 - IT	Learning Outcome Assessment III - IT	3	ASSE 3211 (Senior Level)		Natural Science Elective II**	4	
Total Credit Hours		13		Total Credit Hours		13	
INFORMATION TECHNOLOGY PROGRAM TOTAL DEGREE CREDIT HOURS = 132							

Electives:

*Social Science Electives		**Natural Science Electives	
ECON1311	Intro. to Macroeconomics	BIOL1411	Introductory Biology
ECON1312	Intro. to Microeconomics	CHEM1411	Introductory Chemistry
GEGR1311	World Regional Geography	CHEM1421	Chemistry for Engineers I
HIST1311	World Civilizations	CHEM1422	Chemistry for Engineers II
PSYC1311	Intro. to Psychology	PHYS1411	Introductory Physics
FREN1311	Introduction to French Language		
SPAN1311	Intro. to Spanish Language		
SUST 1311	Introduction to Sustainability		

***IT Electives (Tracks)

	Course Code	Course Title	Pre-requisite
SOFTWARE DEVELOPMENT	ITAP4316	Introduction to Software Project Management	GEIT3351
	ITAP3315	Software Testing & Quality Assurance	GEIT3351
	ITAP4391	Reuse and Component-based Development	GEIT3351
	ITAP3371	Database II	GEIT3341
	ITAP4313	Advanced Visual Programming	ITAP3313, GEIT3341
MOBILE/ WEB	ITAP4313	Advanced Visual Programming	ITAP3313, GEIT3341
	ITAP3362	Mobile Applications Design and Development	ITAP2312, GEIT3341
	ITAP3363	Principles of Open Source & Enterprise Computing with JAVA	ITAP2312, GEIT3341
	ITAP4315	Cloud Computing	GEIT3331, ITAP2431
	ITAP4314	Multimedia Computing and Applications	ITAP2312

	Course Code	Course Title	Pre-requisite
SECURITY	ITAP4361	Operating Systems	ITAP3411
	ITAP4390	Information Security and Assurance	ITAP4361, ITAP3431
	ITAP4396	Computer and Network Forensics	ITAP4390, GEIT3331
	ITAP4376	Secure e-Commerce	GEIT3331, ITAP4371
ADMIN	ITAP4362	Network Administration	ITAP4361, ITAP2431
	ITAP4363	Systems Administration	ITAP4361, ITAP2431
	ITAP4367	Database Administration	ITAP4361, GEIT3341
	ITAP4364	Performance Analysis of Computer Networks	MATH1313, ITAP2431, ITAP3411
	ITAP4393	Special Topics	OVER 90 CR

**** MIS/Business Electives: Any 4000 level courses

B. DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

The Department of Computer Science and Engineering will be administered as a division of the College of Computer Eng. & Science. As such, it will fall under the senior leadership of the Vice Rector of Academic Affairs and the Dean of the College of Computer Eng. & Science.

Responsibility for the operation, administration and management of the department will be vested in the Chair of the Department of Computer Science and Engineering. The chair will oversee the department as a whole, with primary responsibility for instruction of male students. He will delegate responsibilities as appropriate to the Associate Chair, who will oversee instruction for female students.

The Department of Computer Science and Engineering will be charged with the responsibility of maintaining, administering, and managing the degree requirements of the Bachelor of Science in Computer Science and the degree requirements for the Bachelor of Science in Computer Engineering.

1. Content of the Computer Science Degree Program

   The Bachelor of Science in Computer Science is comprised of four components:

   - The PMU Core Curriculum. This core curriculum consists of 60 hours of coursework as defined in the report Undergraduate Core Curriculum Design.
   - Degree-Specific Requirements. These requirements represent support courses in mathematics, laboratory science and business. These requirements both specify and extend Core Curriculum requirements. The degree-specific requirements add 12 credit hours to the degree program.

     The University Core Curriculum requires six semester hours of mathematics. The Computer Science degree program extends that requirement to 18 semester credit hours of mathematics and specifies that the courses will be:

     - MATH 1422: Calculus I
     - MATH 1423: Calculus II
     - MATH 1324: Calculus III
     - MATH 2313: Probability and Statistics
     - MATH 3433: Linear Algebra&Diff. Equations

University Core Curriculum requires eight semester hours of Natural and Physical Science. The Computer Science degree program extends that requirement to 12 semester credit hours of Natural and Physical Science and specifies that the courses will be:

- PHYS 1412 Physics for Engineers I
- PHYS 1413 Physics for Engineers II
- Natural Science elective

- The College of Computer Eng. & Science Requirements. These requirements consist of 29 hours of coursework contained in the nine college courses designated with the GEIT prefix. This coursework represents conceptual and skill-based knowledge that is common to all degree programs within the College of Computer Eng. & Science. (See Section V.B.1., Degree Programs, of this report.)

- The Degree Program Requirements. These requirements consist of coursework as follows.
  - COSC 2312: Web Programming
  - COSC 3332: Discrete Str. & Com. Ana
  - COSC 3351: Algorithms
  - COSC 3361: Computer Networks
  - COSC 3411: System Programming
  - COSC 4361: Operating Systems
  - COSC 4461: Programming Languages
  - COSC 4362: Artificial Intelligence
  - COSC 4363: Theory of Computation

- Electives: The Computer Science Degree Program will require 9 credit hours of elective to be taken from an approved list of 4000 level courses within the College of Computer Eng. & Science.

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

CS - CURRENT DEGREE PLAN

FRESHMAN YEAR							
FIRST SEMESTER				SECOND SEMESTER			
Course Number	Course Title	Credit Hours	Pre-requisite	Course Number	Course Title	Credit Hours	Pre-requisite
ALIS 1211	Introduction to Islamic culture	2		ALIS 1212	The Social System in Islam	2	
COMM 1311	Written Communication	3		COMM 1312	Writing and Research	3	COMM 1311
UNIV 1211	Prof. Development and Competencies	2		UNIV 1212	Critical Thinking and Problem Solving	2	
MATH 1422	Calculus I	4	PRPM 0022	PHYS 1421	Physics for Engineers I	4	PRPM 0022
GEIT 1411	Computer Science I	4		GEIT 1412	Computer Science II	4	GEIT 1411
PHED 1111	Active Living Lifestyle	1		MATH 1423	Calculus II	4	MATH 1422
Total Credit Hours		16		Total Credit Hours		19	

SOPHOMORE YEAR							
FIRST SEMESTER				SECOND SEMESTER			
Course Number	Course Title	Credit Hours	Pre-requisite	Course Number	Course Title	Credit Hours	Pre-requisite
ALIS 2211	Linguistic Communication Skills	2	ALIS 1212	ALIS 2212	The Biography of Prophet Mohammad	2	
UNIV 1213	Leadership and Teamwork	2		COMM 2312	Technical and Professional Communications	3	COMM 2311
PHYS 1422	Physics for Engineers II	4	PHYS1421 MATH 1422	ASSE 2111	Learning Outcome Assessment I	1	Sophomore Level
MATH 1324	Calculus III	3	MATH 1423	GEIT 2331	Mathematical Reasoning and Algorithmic Thinking	3	GEIT 1412
GEIT 2421	Data Structures	4	GEIT 1412	COSC 2312	Web Programing	3	GEIT 1411
COMM 2311	Oral Communication	3	COMM 1312	GEIT 2291	Professional Ethics	2	
PHED 1112	Healthy Behaviors and Management	1		MATH 2313	Probability and Statistics	3	MATH1422
Total Credit Hours		19		Total Credit Hours		17	

JUNIOR YEAR							
FIRST SEMESTER				SECOND SEMESTER			
Course Number	Course Title	Credit Hours	Pre-requisite	Course Number	Course Title	Credit Hours	Pre-requisite
COSC 3332	Discrete Structures and Combinatorial Analysis	3	GEIT 2331	GEIT 3351	Principles of Software Engineering	3	GEIT 1412
GEIT 3341	Database I	3	GEIT 1412	COSC 3361	Computer Networks	3	MATH 2313 GEIT 2421
GEIT 3331	Computer Organization	3	GEIT 1412	COSC 3351	Algorithms I	3	GEIT 2421
MATH 3433	Linear Algebra and Differential Equation	4	MATH 1324	COSC 3411	Systems Programming	4	GEIT 3331
ASSE 3211	Learning Outcome Assessment II	2	ASSE 2111		Natural Science Elective**	4	
	CS Elective I***	3					
Total Credit Hours		18		Total Credit Hours		17	

SUMMER OF JUNIOR YEAR				
Course Number	Course Title	Credit Hours		Pre-requisite
GEIT 4361	Internship	3	8 weeks (320 hours) full time	End of Junior Year (summer before graduation) and department approval

SENIOR YEAR							
FIRST SEMESTER				SECOND SEMESTER			
Course Number	Course Title	Credit Hours	Pre-requisite	Course Number	Course Title	Credit Hours	Pre-requisite
COSC 4361	Operating Systems	3	COSC 3411	COSC 4362	Artificial Intelligence	3	COSC 3351
COSC 4461	Programming Languages	4	COSC 3411	COSC 4363	Theory of Computation	3	COSC 3351 MATH 3433
ASSE 4311 - CS	Assessment III - CS (Capstone)	3	ASSE 3211		Social Science Elective II*	3	
	CS Elective II***	3			CS Elective III***	3	
	Social Science Elective I*	3					
Total Credit Hours		16		Total Credit Hours		12	
TOTAL DEGREE CREDIT HOURS =							

Electives:

*Social Science Electives		**Natural Science Electives	
ECON1311	Intro. to Macroeconomics	**BIOL1411**	Introductory Biology
ECON1312	Intro. to Microeconomics	**CHEM1411**	Introductory Chemistry
GEGR1311	World Regional Geography	**CHEM1421**	Chemistry for Engineers I
HIST1311	World Civilizations	**CHEM1422**	Chemistry for Engineers II
PSYC1311	Intro. to Psychology	**PHYS1411**	Introductory Physics
FREN1311	Introduction to French Language		
SPAN1311	Intro. to Spanish Language		
SUST 1311	Introduction to Sustainability		

***CS Electives

COSC3354	Introduction to Cryptography	GEIT2421
COSC4371	Computer Graphics	GEIT2421
ITAP3313	User Interface Development	GEIT1412, COSC2312
COSC3359	Computer Animation	GEIT2421
COSC3357	Logic and Formal Verification	GEIT2331
COSC4373	Computer Vision	GEIT2421 MATH 2313
COSC4352	Formal Methods in Software Engineering	GEIT3351
COSC4372	Distributed Systems and Algorithms	COSC3351,COSC4361

COSC4376	Bioinformatics	COSC3351
COSC4380	Quantum Information and Computation	COSC3351
COSC4393	Special Topics	OVER 90 CH
COSC4311	Parallel Computing	COSC3351
COSC4364	Compilers	COSC3351, COSC4461
ITAP4371	e-Commerce	GEIT3341, COSC2312
ITAP3371	Database II	GEIT3341

2. Content of the Computer Engineering Degree Program

The Bachelor of Science in Computer Engineering is comprised of five components:

- The PMU Core Curriculum. This core curriculum consists of 60 hours of coursework as defined in the report Undergraduate Core Curriculum Design
- Degree Specific Requirements. These requirements represent support courses in mathematics, laboratory science and business. These requirements both specify and extend Core Curriculum requirements. The degree-specific requirements add 15 credit hours to the degree program.

  The University Core Curriculum requires six semester hours of mathematics. The Computer Engineering degree program extends that requirement to 18 semester credit hours of mathematics and specifies that the courses will be:

  - MATH 1422: Calculus I
  - MATH 1423: Calculus II
  - MATH 1324: Calculus III
  - MATH 2313: Probability and Statistics
  - MATH 3433: Linear Algebra&Diff. Equations

  University Core Curriculum requires eight semester hours of Natural and Physical Science. The Computer Engineering degree program extends that requirement to 12 semester credit hours of Natural and Physical Science and specifies that the courses will be:

  - PHYS 1421: Physics for Engineers I
  - PHYS 1422: Physics for Engineers II
  - Natural Science elective
- The College of Computer Eng. & Science Requirements. These requirements consist of 29 hours of coursework contained in the seven college courses designated with the GEIT prefix. This coursework represents conceptual and skill-based knowledge that is common to all degree programs within the College of Computer Eng. & Science.
- The Degree Program Requirements. These requirements consist of 24 hours of coursework as identified below.
  - COEN 2411: Circuits
  - COEN 3323: Digital and Logic Design
  - COEN 3361: Computer Networks
  - COEN 3421: Electronics
  - COEN 4361: Operating Systems
  - COEN 4322: Digital Signal Processing
  - COEN 4461: Embedded systems
- Electives: The Computer Science Degree Program will require 12 semester credit hours of elective to be taken from an approved list of 4000 level courses within the College of Computer Eng. & Science or the Department of Electrical Engineering in the College of Engineering.

BACHELOR OF SCIENCE IN COMPUTER ENGINEERING

CE - CURRENT DEGREE PLAN

FRESHMAN YEAR

FIRST SEMESTER				SECOND SEMESTER			
Course Number	Course Title	Credit Hours	Pre-requisite	Course Number	Course Title	Credit Hours	Pre-requisite
ALIS 1211	Introduction to Islamic culture	2		ALIS 1212	The Social System in Islam	2	
COMM 1311	Written Communication	3		COMM 1312	Writing and Research	3	COMM 1311
UNIV 1211	Prof. Development and Competencies	2		UNIV 1212	Critical Thinking and Problem Solving	2	
MATH 1422	Calculus I	4	PRPM 0022	PHYS 1421	Physics for Engineers I	4	PRPM 0022
GEIT 1411	Computer Science I	4		GEIT 1412	Computer Science II	4	GEIT 1411
PHED 1111	Active Living Lifestyle	1		MATH 1423	Calculus II	4	MATH 1422
Total Credit Hours		16		Total Credit Hours		19	

SOPHOMORE YEAR

FIRST SEMESTER				SECOND SEMESTER			
Course Number	Course Title	Credit Hours	Pre-requisite	Course Number	Course Title	Credit Hours	Pre-requisite
ALIS 2211	Linguistic Communication Skills	2		ALIS 2212	The Biography of Prophet Mohammad	2	
UNIV 1213	Leadership and Teamwork	2		COMM 2312	Technical and Professional Communications	3	COMM 2311
PHYS 1422	Physics for Engineers II	4	PHYS1421 MATH1422	ASSE 2111	Learning Outcome Assessment I	1	Sophomore Level
MATH 1324	Calculus III	3	MATH 1423	GEIT 2331	Mathematical Reasoning and Algorithmic Thinking	3	GEIT 1412
GEIT 2421	Data Structures	4	GEIT 1412	COEN 2411	Circuits	4	MATH1423 PHYS 1422
COMM 2311	Oral Communication	3	COMM 1312	GEIT 2291	Professional Ethics	`2	
PHED 1112	Healthy Behaviors and Management	1		MATH 2313	Probability and Statistics	3	MATH1422
Total Credit Hours		19		Total Credit Hours		18	

JUNIOR YEAR

FIRST SEMESTER				SECOND SEMESTER			
Course Number	Course Title	Credit Hours	Pre-requisite	Course Number	Course Title	Credit Hours	Pre-requisite
GEIT 3341	Database I	3	GEIT 1412	GEIT 3351	Principles of Software Engineering	3	GEIT 1412
GEIT 3331	Computer Organization	3	GEIT 2421	COEN 3361	Computer Networks	3	MATH 3433 MATH 2313 GEIT 2421
MATH 3433	Linear Algebra and Differential Equation	4	MATH 1324	ASSE 3211	Learning Outcome Assessment II	2	ASSE 2111
COEN 3323	Digital and Logic Design	3	COEN 2411	COEN 3421	Electronics	4	COEN 2411
	CE Elective I***	3			CE Elective II***	3	
Total Credit Hours		16		Total Credit Hours		15	

SUMMER OF JUNIOR YEAR				
Course Number	Course Title	Credit Hours		Pre-requisite
	Internship	3	8 weeks (320 hours) full time	End of Junior Year (summer before graduation) and department approval

SENIOR YEAR							
FIRST SEMESTER				SECOND SEMESTER			
Course Number	Course Title	Credit Hours	Pre-requisite	Course Number	Course Title	Credit Hours	Pre-requisite
COEN 4361	Operating Systems	3	GEIT 3331	COEN 4413	Embedded Systems	4	COEN 4361, GEIT 3351
ASSE 4311 - CE	Assessment III – CE (Capstone)	3	ASSE 3211		Social Science Elective II*	3	
	Natural Science Elective **	4		COEN 4322	Digital Signal Processing	3	COEN 3323 MATH 2313
	CE Elective III***	3			CE Elective IV***	3	
	Social Science Elective I *	3					
Total Credit Hours		16		Total Credit Hours		13	
COMPUTER ENGINEERING PROGRAM TOTAL DEGREE CREDIT HOURS = 135							

lectives:

*Social Science Electives		**Natural Science Electives	
ECON1311	Intro. to Macroeconomics	**BIOL1411**	Introductory Biology
ECON1312	Intro. to Microeconomics	**CHEM1411**	Introductory Chemistry
GEGR1311	World Regional Geography	**CHEM1421**	Chemistry for Engineers I
HIST1311	World Civilizations	**CHEM1422**	Chemistry for Engineers II
PSYC1311	Intro. to Psychology	**PHYS1411**	Introductory Physics
FREN1311	Introduction to French Language		
SPAN1311	Intro. to Spanish Language		
SUST 1311	Introduction to Sustainability		

***CE Electives

Course Code	Course Title	Pre - requisite
COSC3354	Introduction to Cryptography	GEIT2421
COSC4371	Computer Graphics	GEIT2421
COSC4373	Computer Vision	GEIT2421 MATH 2313
COEN4331	Microprocessors	COEN3323 GEIT 3331
COEN4371	Synthesis with Hardware Description Languages	COEN3323

Course Code	Course Title	Pre - requisite
COEN4393	Special Topics	OVER 90 CH
COEN4394	Digital Video Processing	COEN 4322
COEN4395	Wireless and Mobile Computing	COEN 3361
COEN4396	Modern Control Theory	MATH 3433 COEN 3412
COEN4397	Robotics	MATH 3433 COEN 3412

VII. LABORATORY REQUIREMENTS

The College of Computer Eng. & Science will provide laboratory space to accommodate students within the college. As noted in the May 3, 2004 memorandum, PMU Space Program: Report of Zuhair Fayez Architect visit to Austin, Texas, computer labs will typically be built to a size that allocates 2 square meters per student.

Because of the nature of the college, two specific types of physical laboratory space are required

A. GENERAL PURPOSE / COMPUTER LABORATORIES

These laboratories will provide state-of-the-art computing equipment with access to university-wide software resources and to Internet communication. The general-purpose computing laboratories will provide the facilities necessary for many of the classes that include lab components.

1. Lab Design

General-purpose computing laboratories will be designed to accommodate 24 students. This would require providing 24 networked computer systems in each lab. These computer systems should be grouped into pods of four to facilitate group projects. Each pod should have access to scanning and printing facilities. A conceptual design of the laboratory space is shown below.

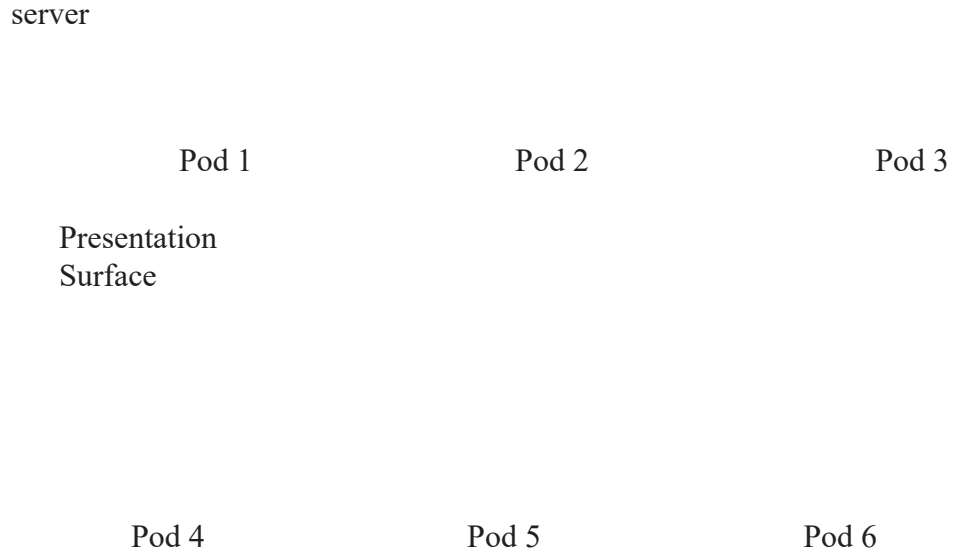

Figure 1: Organizing computer systems in pods facilitates group lab projects.

2. Equipping the Labs

In order to provide students with the kinds of experience necessary for a comprehensive understanding of professional systems, students at the PMU will have access to a wide variety of hardware, operating systems, and professional tools. In order to provide students with the most up-to-date equipment best matched to the needs of their courses, a detailed analysis of the lab needs and hardware/software specifications should be provided by the faculty of the College of Computer Eng. & Science in the months before the PMU's opening. Laboratory resources, however, should incorporate at least one facility that provides dual Windows/Linux boot capability and access to proprietary Unix facilities.

3. Lab Oversight

Responsibility for scheduling the use of the laboratories will reside in the College of Computer Eng. & Science. During times when the laboratories are not scheduled for class use, they should be available to all students in the College of Computer Eng. & Science as open access facilities.

The general-purpose computer laboratories should be under the control and jurisdiction of the College of Computer Eng. & Science. Support for these labs will be provided by the university-wide computer service department under the office of the PMU Chief Information Officer (CIO). This support organization will ensure the maintenance, security and integrity of the hardware. It also will provide supervision services for the facilities during times when they are used as open access resources. (A discussion of the university's support organization as it relates to discipline-specific applications appears in the report Information Technology Strategy.)

B. SPECIAL PURPOSE LABORATORIES

A number of courses within the Information Technology, Computer Science and Computer Engineering degree programs will require specialized equipment and laboratory facilities. The College of Computer Eng. & Science will provide the following in support of those courses:

1. Network Management Laboratory

   The course ITAP 2431: Network Management will require a specialized laboratory in which students may establish and develop network environments. Because of the constructive and experimental nature of this course, the network laboratory must be isolated from the university technology infrastructure. It also cannot easily be used as a general purpose or open access facility. The Network Management Laboratory should consist of six small network pods, each containing three workstations and a server. Each pod should also contain both wired and wireless routing hardware and printing facilities. The College of Computer Eng. & Science through the Department of Information Technology will have control and jurisdiction over this facility.

2. Network Security Laboratory

   The course ITAP 3431: Network Security will require the use of a specialized laboratory in which students may establish and develop techniques for securing network environments. This laboratory must be isolated from the university technology infrastructure and cannot easily be used as a general purpose or open access facility. The Network Management Laboratory facilities cannot easily be leveraged to accommodate this purpose. The Network Security Laboratory should consist of 24 computer systems and printing facilities. The College of Computer Eng. & Science through the office of the university CIO will have control and jurisdiction over this facility.

3. Parallel Computing Laboratory

   The course COSC 4311: Parallel Computing will not require laboratory space. However, the College of Computer Eng. & Science will need to provide parallel computing facilities that may be physically housed in one of the special purpose laboratories but accessed remotely by students during the course. The College of Computer Eng. & Science through the Department of Computer Science and Engineering will have control and jurisdiction over this facility.

4. Digital Signal Processing Laboratory

   The course COEN 4322: Digital Signal Processing will require facilities for the design and implementation of special purpose processors and their application to specific problems. The facility should contain equipment to support the programming and erasing of memory devices and hardware and software for testing and debugging. The facility should consist of 24 computer systems each equipped with DSP boards and software. The College of Computer Eng. & Science through the Department of Computer Science and Engineering will have control and jurisdiction over this facility.

5. Circuits Laboratory

   The course COEN 2111: Circuits I Lab will require facilities for providing hands-on experience working with electrical systems. Courses for male students may use the Circuits Lab facilities of the Department of Electrical Engineering, or the College of Computer Eng. & Science may elect to provide its own lab facilities. Courses for female students will require a Circuits Lab on the female campus of the PMU.

   A Circuits Lab will include the use of analog and digital electronics; communication, computer, and control systems; instrumentation, machinery, and power systems. The laboratory space should be designed as six self-contained areas duplicating the circuits equipment. The College of Computer Eng. & Science through the Department of Computer Science and Engineering will have control and jurisdiction over the Circuits Lab facilities in the IT buildings.

VIII. COURSE SYLLABI

A. COURSE NUMBERING SYSTEM

A common system for naming courses will be applied throughout all academic programs at the PMU. The system is structured as follows:

Each course title begins with four letters that indicate the subject matter of the course. For syllabi in the report Undergraduate Information Technology and Computer Science Programs, these letterings include:

- ASSE Assessment Capstone Series
- GEIT General IT courses
- ITAP Information Technology
- COEN Computer Engineering
- COSC Computer Science

The letters are followed by four numbers:

- First digit indicates the earliest year a course can be taken. A number 1 course may be taken at any time.
- Second digit indicates credit hours. Most courses carry 3 hours of credit. Science courses with labs carry 4 hours of credit. A small number of courses carry 1 or 2 hours of credit.
- Third digit indicates a course that is part of a group or family of courses. For example, calculus courses are assigned the number 2. More advanced math courses are assigned the number 3.
- Fourth digit serves only to differentiate courses from one another within a family. For example, the four calculus courses are numbered 1, 2, 3, and 4.

B. GENERAL INFORMATION TECHNOLOGY COURSES

- GEIT 1411: Computer Science I
- GEIT 1412: Computer Science II
- GEIT 2421: Data Structures
- GEIT 2291: Professional Ethics
- GEIT 2331: Mathematical Reasoning & Algorithmic Thinking
- GEIT 3331: Computer Organization
- GEIT 3341: Database I
- GEIT 3351: Principles of Software Engineering
- GEIT 4361: Practical Training
- ASSE 4311: Learning Assessment III

GEIT 1411: Computer Science I

Semester Credit Hours: 4 (3,1)

I. Course Overview

Computer Science I is an introduction to programming and to the use of algorithms in designing programs. A software engineering approach to developing computer programs is stressed and object-oriented concepts are introduced. The course examines standard control structures, approaches to modularization, and the use of primitive and structured data types.

II. PMU Competencies and Learning Outcomes

Students of GEIT 1411 will develop both the conceptual basis and the practical skills in the design and implementation of programs to solve specific problems. These fundamental skills are necessary for continued success in Computer Engineering, Computer Science and information Technology. This course will make extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course includes a structured laboratory component to ensure that students gain the necessary experience and skill in handling Application Development Environments. The course will include individual as well as group projects and provide opportunities for the presentation and defense of designed solutions.

III. Detailed Course Description

GEIT 1411 is an introductory course in the design and implementation of solutions to specific problems and an introduction to computer programming languages and to the structures that such languages contain. In addition, the course provides experience in the management of application development environments. The course examines the organization of computer programs and the control structured used to manage the flow of instructions. Primitive data types are introduced along with techniques for manipulating those data types and their use in building aggregate data structures such as strings, and arrays. Techniques for improving readability and maintainability are taught including appropriate documentation techniques and program modularity, initially through functions, and later though object oriented techniques.

IV. Requirements Fulfilled

GEIT 1411 satisfies 4 hours of the requirements for degrees in Computer Engineering, Computer Science and information Technology. It is required of all students pursuing a degree program within the College of Computer Engineering and Science. It should be taken in the first semester of the freshman year.

V. Required Prerequisites

None. This is first course in the Computer Engineering, Computer Science and Information Technology degree programs.

VI. Learning Outcomes

By the end of the course the students should be able to:

CLO1. Differentiate and convert between various data types and apply the appropriate ones in programming statements

CLO2. Analyze problems, construct and implement solutions using selective, iterative and sequential statements

CLO3. Utilize functions (user defined and predefined) to achieve programming goals.

CLO4. Use arrays and strings and solve problems efficiently using loops

CLO5. Use application development environments as professional tools in program development (e.g., Netbeans) to write, test and debug programs

VII. Assessment Strategy

This course is designed with three primary goals in mind; to introduce students to the conceptual basis and practical issues associated with the development of computer programs, to provide students with significant experience in the development of computer programs within a profession development environment, and to provide students with the opportunity to communicate their designs and implementations to their peers in a professional setting. With this in mind, the course grade involves an assessment of their performance on exams that focus on the application of programming techniques to the solution of problems and the communication of designed solutions to those problems to an audience. Course grades are be based on

- Weekly assigned homework to motivate students to do the work and earn credit accordingly.
- Weekly, in-class presentations by students of solutions to real world problems related to the course material and classroom discussion and critique of the presentation.
- Weekly structured laboratory exercises designed to guide students through specific course topics.
- Two in-class exams to assess students' accumulative mastery of content covered prior to time of exam.
- 8 programming assignments testing students understanding of the major concepts introduced during the course
- A comprehensive final exam to assess students' accumulative mastery of course material.

Students' final grades will be based on 10% credit for the homework, 10% for the presentations and participation in classroom discussion, 20% for weekly lab exercises, 20% on in-class exams, 30% on programming assignments and 10% for the final examination.

Students are required to maintain a journal of thoughts and commentaries during the course. The journal will contain daily entries including the identification of areas of interest and concern, notes on the preparation of presentation and comments and analysis of classmate's presentations. The journal will be reviewed weekly by the instructor to provide feedback to the students.

Final grades and the student and instructor observations from reflective notebooks will be included in the student's portfolio for use in the final assessment capstone course. The intent is to document

the student's maturation as he proceeds through the curriculum.

VIII. Course format

Instruction: This course utilizes both lecture/discussion and laboratory exercises. Students are expected to attend three hours of lecture/discussion per week and two hours of laboratory per week. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Once a week students should have at least 30 minutes of collaborative problem solving activity.

- Course syllabus.
- Course assignments.
- Keys to exams (after students have completed them).
- Model programmed solutions to programming assignments (once students have completed them)
- Course calendar (an active utility).
- Course e-mail (an active utility).
- Course discussion list (an active utility).
- Students course marks. (an active utility).

Classroom Hours (5 hours per week) Class: 3
Lab: 2

Web supplement: Course home page (the university's Web tool, WebCt or Blackboard) should contain the following:

IX. Topics to be covered

The course will cover the following topics:

1. Introduction
    a. Programming and Problem Solving
    b. Introduction to the programming language
    c. Testing and Debugging
2. Introduction to Java Application:
    a. Variables and assignment
    b. Input/Output
    c. Data Types, Operators and Expressions
    d. Flow Control
3. Introduction to classes, Objects, Methods and String
    a. Instance Variable
    b. Set and Get Method
    c. Primitive type vs Reference type
    d. Objects with constructor
4. Control Statements: part 1
    a. Algorithm, Pseudocode
    b. If – Single Selection
    c. If…else- Double Selection
    d. Nested if
    e. While repetitions
5. Control Statement: part 2
    a. Essentials of Counter controlled repetitions
    b. For repetition statement
    c. Do while repetition statement
    d. Switch multi selection statement
    e. Break and continue
6. Methods: A deeper look
    a. Static methods, fields, class and Math
    b. Declaring methods with multiple parameter
    c. Declaring and using methods
    d. Argument promotion and casting
    e. Scope of declaration
    f. Method overloading
7. Arrays and ArrayLists
    a. Declaring and creating Arrays
    b. Enhanced for statement
    c. Passing Arrays to Methods
    d. Pass By Value vs Pass By Reference
    e. Multidimensional Array
    f. Two-Dimensional Array
    g. Class Array

XI. Technology component

This course makes use of the university's wireless access infrastructure. The course relies on the university and the students having access to professional grade application development environments for the students to use. The course has a laboratory component which would be best implemented in university provided laboratory space.

XII. Special Projects / Activities

Students are required to keep a "reflective notebook" in which, after each class, they enter their own assessments of what they learned, and what questions remain from the class. From each exercise set, each student will select one problem, which the student thinks best reflects the way the mathematical topic will be used in a technical context. A detailed solution to the problem will be included in the student's reflective notebook.

XIII. Textbooks and Teaching aids

A. Required Textbook
JAVA- how to program early objects, 10th edition, Dietel & Dietel.

B. Instructors Materials on Blackboard

GEIT 1412: Computer Science II

Semester Credit Hours: 4 (3,1)

I. Course Overview

This course is a continuation and extension to GEIT 1411 Computer Science I. It introduces the student to a systematic study of basic data structures such as linked lists. A software engineering approach to developing computer programs is stressed and object-oriented concepts are emphasized. Reusability of code, effective software development methodologies and good programming practices are significant components of the course.

II. PMU Competencies and Learning Outcomes

Students of GEIT 1412 extend and develop skills introduced in GEIT 1411, Computer Science I. These skills are necessary for professional success in computer engineering, computer science and information technology. This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course includes a structured laboratory component to ensure that students gain the necessary experience and skill in managing the concepts introduced in the class. The course includes individual as well as group projects and provides opportunities for the presentation and defense of designed solutions.

III. Detailed Course Description

As a continuation of GEIT 1411: Computer Science I, GEIT 1412: Computer Science II presents students with the concepts, implementation and characteristics of basic, well-known data structures such as linked lists. An object-oriented approach to development is stressed to encourage code reuse, encapsulation and to illustrate concepts such as inheritance, polymorphism and interfaces. Students are exposed to a variety of techniques for the implementation of each data structure including static and dynamic data structures and recursive solutions. One important lasting effect of this course is to enhance and develop the ability to specify, design, implement and test solutions to programming problems utilizing the data structures and proven algorithms presented in this course.

IV. Requirements Fulfilled

This course is required for all students regardless of major in Information Technology in the College of Computer Engineering and Science. It should be taken in the second semester of the freshman year.

V. Required Prerequisites

GEIT 1411: Computer Science I

Student also are expected to have successfully completed the first mathematics course in their degree program.

VI. Learning Outcomes

In this course, students learn:

1. List, define and contrast basic object oriented concepts, including classes and public/private membership
2. Develop the necessary skills to apply file i/o with advanced data types
3. Define linked lists and recursion and utilize as efficient solutions to specific problems
4. Explain advantages and disadvantages of advanced object oriented constructs including Inheritance, Polymorphism and Interfaces
5. Develop the communication, leadership and teamwork skills necessary to work in or lead of teams

VII. Assessment Strategy

This course is designed with three primary goals in mind: to introduce students to the conceptual basis and practical issues associated with the development of computer programs utilizing basic well known data structures, to provide students with significant experience in the development of computer programs from an object-oriented perspective within a profession development environment, and to provide students with the opportunity to communicate their designs and implementations to their peers in a professional setting. With this in mind, the course grade involves an assessment of their performance on exams that focus on the application of programming techniques to the solution of problems and the communication of designed solutions to those problems to an audience. Course grades are be based on:

- Project to motivate students to collaborate and earn credit accordingly.
- Weekly structured laboratory exercises designed to guide students through specific course topics.
- Quizzes and a midterm exam to assess the student's accumulative mastery of content covered prior to time of exam.
- programming assignments testing the student's understanding of the major concepts introduced during the course
- A comprehensive final exam to assess the student's accumulative mastery of course material.

•

Assignments	5 %	
Project	10 %	
Midterm Exam		20 %
Lab	20 %	
Quizzes	15 %	
Class participation	5 %	
Final Exam	25 %	
Total	100 %	

Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he or she proceeds through the curriculum.

VIII. Course Format

This course utilizes both lecture/discussion and laboratory exercises. Students are expected to attend three hours of lecture/discussion per week and two hours of laboratory per week. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Once a week students should have at least 30 minutes of collaborative problem solving activity.

Classroom Hours (5 hours per week) Class: 3
Lab: 2

Web supplement: Course home page (the university's Web tool, WebCT or BLACKBOARD) should contain the following:

- Course syllabus
- Course assignments
- Sample solutions to examinations (after being graded and returned)
- Sample solutions to programming assignments (after being graded and returned)
- Course calendar (an active utility)
- Course e-mail (an active utility)
- Course discussion list (an active utility)
- Student course performance (an active utility)

IX. Topics to be covered

A. Object-oriented design

1. Goals and principles
2. Class design issues
3. Inheritance and polymorphism
4. Interfaces

B. Advance concepts

1. Exceptions
2. Recursion
3. I/O using Files and Streams

C. Lists

1. Introduction to the linked list ADT
2. Array-based implementation of the linked list

D. GUI

1. Basic graphical user interface

X. Laboratory Exercises

This course requires a weekly two-hour lab component. Topics to be covered in the laboratory sessions should include:

- Class and objects
- Inheritance (it covers sub and super class concepts)
- Polymorphism and interfaces
- Exception handling
- Files and Streams
- Linked list (it covers how to create linked list manually, without using Generic Collections)
- Recursion
- GUI introduction

XI. Project samples

- Design of social network engine using two dimensional linked list.
- Design of university database using two dimensional linked list.
- Design of Vehicle database system using three dimensional linked list.
- Design of Email Inbox system using single dimensional linked list.
- Etc.

XII. Technology Component

This course makes use of the university's wireless access infrastructure. The course relies on the university and the students having access to professional grade application development environments for the students to use. The course has a laboratory component that would be best implemented in university-provided laboratory space.

XIII. Special Projects/Activities

Students are required to keep a "reflective notebook" in which, after each class, they enter their own

assessments of what they learned, and what questions remain from the class. From each exercise set, each student selects one problem, which the student thinks best reflects the way the topic is used in a technical context. A detailed solution to the problem is included in the student's reflective notebook.

XVI. Textbooks and Teaching Aids

A. Required Textbook

Paul Deitel, Harvey Deitel, Java How to Program Early Objects, (10th Edition, 2015) Pearson

D. Supplemental Online Materials

As available from publisher.

GEIT 2421 Data Structures

Semester Credit hours: 4 (3, 2)

I. Course overview

Data structures are the systematic study of some advanced data structures, including list, stack, queue, dictionary, and graph. Sorting and hashing algorithms and their associated computational costs are discussed. Algorithm analysis techniques are also investigated to provide a metric to measure the performance of an algorithm in question.

II. PMU Competencies and Learning outcomes

Students in this course develop programming and quantitative skills necessary for continued success in computer science. The skills enhance their abilities to devise, analyze, and comprehend mathematically the performance characteristics of algorithms and data structures common to computer science as a discipline and to effectively communicate their solutions to fellow professionals. This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course includes a structured laboratory component to ensure that students gain the necessary experience and skill in managing the concepts introduced in the class. The course includes individual as well as group projects, establishes both mathematical reasoning skills and technical communication skills, and provides opportunities for the presentation and defense of designed solutions.

III. Detailed course description

GEIT 2421: Data Structures is concerned with the systematic study of some advanced data structures, including list, stack, queues, dictionary, graphs. Sorting and hashing algorithms and their associated computational costs are discussed. The course presents the students with the concepts of asymptotic notations, performance measurement, sorting and searching including algorithms and lower bounds, abstract data types and classes, data structures such as heaps, search trees, tries, and hashing, and graphs: representation, depth-first-search, and breadth-first search. One important lasting effect of this course is to enhance and develop the ability to specify, design, implement, test, and analyze solutions to programming problems utilizing the data structures and proven algorithms presented in this course.

IV. Requirements Fulfilled

This course satisfies four hours of the requirements for the degree in computer science. It is required of all students pursuing a degree program in computer science within the College of Computer Engineering and Science. It should be taken in the first semester of the junior year.

V. Required Prerequisites

GEIT 1412: Computer Science II

VI. Learning Outcomes

In this course, students learn:

- To describe the usage of various data structures. These include lists, stacks, queues, dictionaries, and graphs.
- To describe the usage of various data structures algorithms such as Sorting and Hashing.
- To analyze the performance characteristics of algorithms using mathematical and measurement techniques.
- To explain and summarize the advantages and disadvantages of various data structures implementations.
- To Design and apply appropriate data structures for solving computing problems
- To develop improved communication and collaborative skills.

VII. Assessment Strategy

This course is designed with three primary goals in mind: to introduce students to the conceptual basis and practical issues associated with the use, development, and analysis of data structures, to lead students to connect the mathematics to its application in computer science, and to provide students with the opportunity to communicate their designs and implementations to their peers in a professional setting. With this in mind, the course grade involves an assessment of their performance on examinations that focus on the application of programming techniques to the solutions of problems, the performance analysis of the designed solutions, and the communication of designed solutions to those problems to an audience. Course grades are based on:

Weekly assigned homework to motivate students to do the work and earn credit accordingly.

Weekly, in-class presentations by students of solutions to real world problems related to the course material and classroom discussion and critique of the presentation.

Weekly structured laboratory exercises designed to guide students through specific course topics.

Two in-class examinations to assess the student's accumulative mastery of content covered prior to the time of the examination.

Five programming assignments testing students understanding of the major concepts introduced during the course.

A comprehensive final examination to assess the student's accumulative mastery of course material.

HW's	10%
Midterm 1	20%
Midterm 2	20%
Final Exam	30%
Lab	20%
Total	100%

VIII. Course Format

A. Instruction

This course utilizes both lecture/discussion and laboratory exercises. Students are expected to attend three hours of lecture/discussion per week and three hours of laboratory per week. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Once a week students should have at least 30 minutes of collaborative problem solving activity.

B. Web supplement

Course home page (the university's Web tool, Blackboard) should contain the following:
Course syllabus
Course assignments
Sample solutions to examinations (after being graded and returned)
Sample solutions to programming assignments (after being graded and returned)
Course calendar (an active utility)
Course e-mail (an active utility)
Course discussion list (an active utility)
Student course performance (an active utility)

C. Classroom Hours:

Class: 3 hours per week
Lab: 2 hours per week

D. Topics to be Covered

A. Basic data structures
1. List

2. Stack
3. Queue

B. Analysis of algorithms
1. Time complexity
2. Upper and lower bounds

C. Sorting
1. Insertion sort
2. Merge sort
3. Quick sort
4. Heap sort
5. Performance bounds

D. Dictionary
1. Binary search trees
2. AVL tree
3. B tree

E. Graph
1. Definition, representation, and modeling tool
2. Breath-first search
3. Depth-first search

4. Directed graph and topological sort

F. Hashing
1. Hash tables and functions
2. Collision resolution

IX. Laboratory Exercises

This course requires a weekly two-hour laboratory component. Topics to be covered in the laboratory sessions should include:

Basic data structures – review exercises in basic data structures.

Time complexity – exercises in estimating time complexity of algorithms by measurement.

Performance bounds – exercises in estimating performance bounds of algorithms by measurement.

Sorting I – exercises in the use and analyzing of sorting algorithms including merge sort and quick sort.

Sorting II – additional exercises in the use and analyzing of sorting algorithms including heap sort.

Dictionary I – exercises in the use of dictionary as an abstract data type including binary search tree and AVL tree.

Dictionary II – additional exercises in the use of dictionary including B tree.

Breath-first search – exercises in the use of breath-first search algorithm.

Depth-first search – exercises in the use of depth-first search algorithm.

Topological sort – exercises in the use of topological sort algorithm

Hashing I – exercises in the implementation of hash tables and functions.

Hashing II – exercises in the implementation of collision resolution.

Three additional lab sessions should be kept in reserve to allow the instructor to extend the more difficult laboratories for more than one session.

Technology Component

This course makes use of the university's wireless access infrastructure. The course relies on the university and the students having access to professional grade application development environments for the students to use. The course has a laboratory component that would be best implemented in university provided laboratory space.

Special Projects/Activities

Students are required to keep a "reflective notebook" in which, after each class, they enter their own assessments of what they learned, and what questions remain from the class. From each exercise set, each student selects one problem, which the student thinks best reflects the way the topic is used in a technical context. A detailed solution to the problem is included in the student's reflective notebook.

E. Textbooks and Teaching Aids

Required Textbook

"Java Software Structures: Designing and Using Data Structures", fourth edition Pearson, 2013.

ISBN 9781119186526
By:John Lewis, Joseph Chase.

Alternative Textbooks

Data Structures: Abstraction and Design Using Java, 3rd Edition, January 2016
Elliot B. Koffman, Paul A. T. Wolfgang

F. Class RULES

A. Disruptive Behavior

Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal. Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

B. Class attendance

Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

C. Tardiness

When a student is late for 3 times it is counted as one absent.

G. WITHDRAWAL

It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

GEIT 2291: Professional Ethics

Semester Credit Hours: 2 (2,0)

I. Course Overview

This course is designed to educate students on the impact ethical issues have on the use of information technology in the modern business world. It examines the ethical codes of the professional societies and the philosophical bases of ethical decision-making. Students acquire the foundation that helps them make appropriate decisions when faced with ethical dilemmas.

II. PMU Competencies and Learning Outcomes

This course introduces students to the importance of ethics and professional conduct in their profession. Students develop both the conceptual basis and the practical skills in solving various ethics-related problems arising in various aspects of a modern organization. Students learn to distinguish between ethical and unethical conduct as well as take corrective actions to prevent or solve unethical situations. More importantly, discussion of cases in ethics and professional conduct involves students in the critical thinking process using ethical principles and IT concepts. This course makes an extensive use of the PMU technology infrastructure to provide communication between faculty and students. A significant part of this course involves on-line discussion of cases in ethics and professional conduct.

III. Detailed Course Description

It is critical that students of information technology are introduced to the impact that ethical issues have on the use of information technology in the modern business world. Through an examination of ethical codes of various professional societies, the philosophical bases of ethical decision-making, and several contemporary case studies, students develop the foundation that prepares them to make appropriate decisions when faced with ethical dilemmas. They learn the five rules of ethical behavior, examine how personal values influence professional behavior, and follow a ten-step process for solving ethics-related business problems. In short, this course helps students recognize and think through ethical issues when they arise, correct unethical practices that may have been previously unnoticed or ignored, and communicate the need for applying ethical principles at all organizational levels.

A significant portion of this course is devoted to the discussion of cases in ethics. Two types of case studies are used .The "What would you do?" scenarios present real-to-life dilemmas that can be used as a basis for student exercises to involve student in the critical thinking process using principles presented in class. Real-world cases reinforce important ethical principles and IT concepts and illustrate how real companies have addressed ethical issues associated with IT.

IV. Requirements Fulfilled

This course is required for all students regardless of major in Information Technology in the College of Computer Engineering and Science. It should be taken in the first semester of the sophomore year.

V. Required Prerequisites

This course does not have a prerequisite.

VI. Learning Outcomes

In this course, students learn:

- To describe the importance of ethics and professional conduct in the workplace
- To recognize and prevent unethical behavior as well as promote ethical behavior in the workplace.
- To describe the code of ethics and professional conduct of various professional organizations such as IEEE ACM, and AITP
- To explain how codes of ethics, professional organizations, certification, and licensing affect the ethical behavior of IT professional
- To describe the key trade-offs and ethical issues associated with the safeguarding of data and information systems, software engineering, and software quality.
- To develop in students the notion of accountability in their profession as well as to society at large

VII. Assessment Strategy

The student's performance in this course is assessed on the basis of:

- One mid-term and one final examination.
- Four out-of-class assignments in the form of mini-cases on professional ethics and conduct.
- Four mini cases to be discussed in an online, discussion group format.

Relative weights assigned to these items in determining student's final grade are suggested as follows:

- Each of the two examinations accounts for 30% of the final grade. Combined, the two examinations account for 60% of the grade.
- Four mini-cases submitted in written form account for 20% of the grade.
- Four On-line cases account for 20% of the grade.

The two essay-based examinations are used to assess an understanding of ethical theories and concepts covered in class. Cases based on "What would you do?" scenarios are used to assess student's understanding of ethical principles and process for solving ethics-related problems. Real-world cases are used to assess student's understanding of how real companies have addressed ethical issues associated with IT.

VIII. Course Format

This course utilizes a mix of in-class lectures and student- as well as instructor-led discussions designed to help students develop a deep understanding of the role of ethics and professional conduct in their chosen discipline. While class meetings are utilized to introduce students to the theory of ethical reasoning and process for resolving ethical conflicts, contemporary case studies and "What would you do?" scenarios are used to impart an understanding of guiding principles for ethical and professional conduct. These cases are drawn from the required textbook and supplementary printed material.

Once every two weeks, the instructor assigns one real-world case for students to analyze and respond with a written report. Randomly selected students are asked to make oral presentations of their response.

Once every two weeks and alternating with written case analyses, the instructor posts a "What would you do?" scenario for on-line discussion. An online discussion group is set up to discuss these scenarios outside of the classroom. Students are required to actively participate in this online forum. In addition to contributing to the discussion, the instructor monitors discussion activity for grading purposes. Further, the instructor chooses students at random to present an oral summary of the discussion to the class. This motivates all students to participate in on-line case discussions.

In addition, the instructor should consider creating a Web site for this course using Web technologies such as WebCT or BLACKBOARD. At minimum, the site should include:

- Course syllabus
- Lecture material (for example, PowerPoint slides, lecture notes, etc.). These should be placed on the site ahead of class meeting so that students may use the material to prepare for the lecture
- Summary of cases in ethics that are discussed in class
- Out-of-class assignments in the form of mini cases in Ethics
- Keys to exams (after students have completed them)
- Suggested solution to mini cases (after graded assignments have been returned)
- Mechanism for students to digitally submit their assignments
- Course calendar
- Mechanism to communicate electronically (for example, e-mail)
- Discussion groups. Students participate in one on-line case discussion every two weeks.
- The student's course performance measures

Classroom Hours (2 hours per week) Class: 2
Lab: 0

IX. Topics to be covered

A. Overview of ethics
   1. Why ethics
   2. Ethics in the business world
   3. Forces that shape ethical behavior
   4. Introduction to ethical reasoning
   5. Solving ethical problems - The Ten Step method

B. Ethics for IT professionals and IT users
   1. Ethical behavior of IT professionals
   2. Code of ethics and professional conduct for Association for Computing Machinery, Association of Information Technology Professionals, Software Engineering (IEEE-CS/ACM), and IEEE.

C. Cybercrime
   1. Types of attacks
   2. Risk assessment
   3. Prevention, detection, and response

D. Information and personal privacy
   1. Invasion of privacy
   2. Electronically sharing/selling personal information
   3. Consumer profiling
   4. Spamming

E. Intellectual Property
   1. What is intellectual property?
   2. Key intellectual property issues (copyright infringement, software piracy, etc.)

F. Software Development
   1. Importance of software quality
   2. Software development process

G. Employer/Employee Issues
   1. Use of non-traditional workers (contract, off-shoring)
   2. Ethical considerations in interacting with supervisors, subordinates, and peers
   3. Ethics and performance appraisal

X. Laboratory Exercises

This course does not require a separate lab.

XI. Technology Component

- In class, the instructor makes use of state-of-the art multimedia projection equipment and software. These are used to project slides and Web-based content relevant to the concepts of and issues in professional ethics.
- Outside class, the instructor uses Web-based course management software (for example, WebCT, BLACKBOARD) to interact with students as described under course format section.
- All case analyses are submitted and examinations are taken online. Further, an online discussion group is set up to discuss some of the case studies assigned.

XII. Special Projects/Activities

There are no special projects or activities assigned in this course.

XIII. Textbooks and Teaching Aids

A. Required Textbook

Reynolds, George, Ethics in Information Technology, 5th Edition, Cengage Learning US, 2015. ISBN 978-1285197159

B. Alternative Textbooks

None.

C. Supplemental Print Materials

Maddux, Robert B., and Dorothy Maddux, Ethics in Business: Manage with Authority and Fairness, 2nd Edition, Axzo Press, 2010. ISBN: 978-1426019388

D. Supplemental Online Materials

As available from publisher.

GEIT 2331: Mathematical Reasoning & Algorithmic Thinking

Prerequisites

MATH 1423: Calculus II

Credit hours

3 Credit Hours

Course overview

Discrete Structures, in general, is the study of objects that have discrete as opposed to continuous values including the foundations of logic, algorithms and their complexity, mathematical reasoning, relations, graphs, trees and combinatorics. GEIT 2331: Mathematical Reasoning and Algorithmic Thinking is a mandatory course for all students in the College of Computer Engineering and Science and as such, its goal is to provide students with logical reasoning and other basic mathematical skills that will help them in subsequent courses in their programs and their future careers.

PMU Competencies and Learning outcomes

Students of GEIT 2331: Mathematical Reasoning and Algorithmic Thinking develop the quantitative skills necessary for continued success in computer science. These skills enhance their ability to both analyze and describe mathematically many of the algorithms and data structure performance characteristics common to computer science as a discipline and to effectively communicate their solutions to fellow professionals. The course makes extensive use of the technology infrastructure of the school for communication within the class and between students and the instructor. Through the use of group tasks and projects this course establishes both mathematical reasoning skills and technical communication skills.

Detailed course description

GEIT 2331: Mathematical Reasoning and Algorithmic Thinking is concerned with the application of objects with discrete characteristics to computer science as a discipline in order that commonly used structures may be described, characterized and analyzed. The course examines the fundamentals of propositional and predicate logic, set operations, functions, sequences and summations, matrices, design and analysis of algorithms, mathematical reasoning including proofs and induction, recursion, and program correctness.

Learning Outcomes

In this course, students learn to:
1. Define and apply various discrete structures concepts including sets, functions, sequences, summations and counting techniques.
2. Use mathematical reasoning techniques including proofs, induction and recursion to solve computing problems.
3. Design, analyze and apply simple algorithms, with a focus on integers and related operations.
4. Communicate the solutions of technical problems to other professionals.

Required Text

Required Textbook

Rosen, K. H. (2013) Discrete Mathematics and Its Applications. 7th Edition, McGraw-Hill.
ISBN 978-0-07-131501-2

Alternative Textbooks

Richard Johnsonbaugh. (2005) Discrete Mathematics,
Sixth Edition, Prentice Hall.
ISBN 0-13-117686-2

Hall, C., & O'Donnell J. (2000) Discrete Mathematics Using a Computer, Springer Verlag
ISBN 1-85-233089-9
Balakrishnan, V.K. (1996). Introductory Discrete Mathematics, Dover
ISBN 0-48-669115-2

class RULES

Disruptive Behavior

Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance

Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness

When a student is late for 3 times it is counted as one absent.

WITHDRAWAL

It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment

Grades shall be computed on the basis of the following:

Class participation	5%
Homeworks	15%
Quizzes	20%
Exam I	15%
Exam II	15%
Final Exam	30%
Total	100%

Course Format

A. Instruction

Primary instruction is a lecture format, with the course meeting for three hours per week.

B. Web supplement

Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:

Course syllabus

Lecture material (PowerPoint slides, lecture notes, etc.). These will be placed on the site ahead of class meeting so that students may use the material to prepare for the lecture.
Course assignments
Sample solutions to examinations (after being graded and returned)

CLASS SCHEDULE (Tentative)

Week#	Topics covered	Sections to Read	Practice Problems
1	Introduction Propositional Logic	1.1	6, 10, 15, 18, 20, 26.
2	Applications of Propositional Logic Propositional Equivalences	1.2 1.3	4, 6, 10, 14, 20, 24. 4, 8, 12, 18, 35, 36.
3	Predicates and Quantifiers Nested Quantifiers	1.4 1.5	6, 16, 21, 26, 32, 36. 4, 10, 19, 24.
4	Rules of Inference	1.6	4, 8, 12, 15, 19.
5	Proof Methods	1.8 1.9	
6	Sets Set Operations	2.1 2.2	
7	Functions	2.3	
8	Sequences and Summations	2.4	
9	Cardinality of Sets Matrices	2.5 2.6	
10	Algorithms	3.1	
11	Growth of Functions	3.2	
12	Complexity of Algorithms	3.3	
13	Divisibility and Modular Arithmetic	4.1	
14	Integer Representations and Algorithms	4.2	
15	Mathematical Induction	5.1	

GEIT 3341: Database I

Prerequisites

GEIT 1411: Computer Science I
GEIT 1412: Computer Science II.

Credit hours

3 Credit Hours

Course overview

The objective of this course is to give students an understanding of key issues related to database design and implementation to support the automation of key business processes in organizations. The course is designed so as to cover topics that are relevant from a database design and implementation perspective.

PMU Competencies and Learning outcomes

This course helps students develop the ability to become conversant with applied database design issues and understand the related terms and issues that are important for database designers around the world. Additionally, the course provides the students with the communication, leadership and teamwork skills necessary to effectively work as professionals in teams, or in charge of teams, responsible for database design projects. Finally, the course imparts on the students an understanding of database design as more than simply the design of data repositories, but also as the design of database management resources that support the core and mission-critical business processes of an organization.

Detailed course description

The course begins with a discussion of ethical issues, legal issues, and aspects conducive to effective teamwork, in the context of database design. It then proceeds with a review of concepts and methods that are applicable to most database design projects. This covers concepts such as those of records and fields to diagramming tools such as entity-relationship diagrams. Next the course covers a central topic in database design, namely normalization. The 1st, 2nd and 3rd normal forms are covered in detail. Higher-level normal forms are also covered, although more briefly. The course concludes with a discussion of advanced issues in connection with database design, geared at Web-based access to single and multiple databases. The emphasis in this course is on database design using CASE tools, which allow designers to minimize the amount of low-level programming they have to perform in order to develop functional database systems.

Learning Outcomes

In this course, students learn:

- Define and Diagram entity relationship model and relate to relational databases
- Describe, contrast and apply Relational Model and Relational Algebra
- Justify, Contrast and Apply the various levels of Normalization
- Download, Install and configure a professional database system, e.g., Oracle or MySQL
- Develop the communication, leadership and teamwork skills necessary to work in, or in charge of teams
- Define, contrast and apply DDL and DML SQL

Required Text

Required Textbook
Thomas M. Connolly and Carolyn E. Begg; Database Systems: A Practical Approach to Design, Implementation and Management; 4th edition (May 24, 2004) Pearson Addison Wesley
ISBN: 0321210255.

Class RULES

Disruptive Behavior

Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance

Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non-attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness

When a student is late for 3 times it is counted as one absent.

WITHDRAWAL

It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment

Students are assessed based on: their performance in two exams (midterm and final); their class participation, which includes programming assignment, discussion of recent articles taken from online industry publications; and the quality of a final team project and related oral presentation. The relative weights of each of these items on the final grade are as follows:

GRADING SCHEME:

Participation	5%
Quizzes	5%
Written Assignments	10%
Practical Assignments	20%
Midterm	20%
Project	20%
Final	20%
Total	100%

Course Format

A. Instruction

Four of the course's class meetings are used for laboratory demonstrations and activities geared at helping the students learn the several steps involved in designing and implementing a database system. The other class meetings are split into two main components: lectures, and class discussions. The lectures cover several topics outlined later in this syllabus. The class discussions are based on recent articles taken from online industry publications such as the Searchers and CIO magazines, which are freely available from the Web. The instructor provides the links to the articles, which are then downloaded by the

students and read prior to class. In class, the students discuss the articles in small teams for about 20 minutes, developing three provocative questions per team. This is followed by a discussion involving the whole class, where each team asks one of the questions they developed, and other teams answer them, until all teams asked at least one of their questions. This discussion format is likely to lead to lively debate on topics that are directly addressed by the article, as well as on topics that are indirectly related to the article.

B. Web supplement

Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:
Course syllabus
Course assignments
Sample solutions to examinations (after being graded and returned)
Sample solutions to programming assignments (after being graded and returned)
Course calendar (an active utility)
Course e-mail (an active utility)
Course discussion list (an active utility)
Student course performance (an active utility)

CLASS SCHEDULE (Tentative)

Week	Chapters	Topics
1	Chapter 1	Introduction to Databases
2	Chapter 2	Database Environment
3	Chapter 3	Database Architecture and the Web
4	Chapter 4	The Relational Model
5	Chapter 5	Relational Algebra and Relational Calculus
6	Chapter 6	SQL: Data Manipulation
7	MIDTERM	
8	Mid Semester Break	March 22 - 26
9	Chapter 6	SQL: Data Manipulation
10	Chapter 7	SQL: Data Definition
11	Chapter 8	Advanced SQL – Stored Procedures
12	Chapter 8	Advanced SQL - Triggers
13	Chapter 12	Entity-Relationship Modeling
14	Chapter 13	Enhanced Entity-Relationship Modeling
15	Chapter14	Notmalization
16	Final Exam	

GEIT 3331: Computer Organization

Semester Credit Hours: 3 (3,0)

I. Course Overview

This course examines the functional components of computer systems. Topics discussed include processors, memory types and hierarchies, buses, I/O, interrupts, etc. with emphasis on how they affect program execution, parameter passing and inter-program communications between programs written in diverse languages.

II. PMU Competencies and Learning Outcomes

Students in this course develop a conceptual understanding of the internal structures and mechanism in computer systems. In doing so, students gain an insight into the interaction between software and hardware that allows a more sophisticated understanding of the nature of software and the systems upon which that software executes.

The course provides opportunities for technical skill development as well as communication, collaboration and leadership skills through the maintenance of journals, detailing progress in group projects, and in-class presentations.

This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course includes individual as well as group projects and establishes both conceptual reasoning skills and technical communication skills.

III. Detailed Course Description

GEIT 1311: Computer Organization examines the structures and components common to all commercial computer systems and the fundamental techniques used to effectively and efficiently handling processing tasks. This course reflects the current state of the field, as well as introducing the principles that are shaping the design of computing environments. Students in every specialty of computing need to appreciate the organizational paradigms that determine the capabilities, performance, and, ultimately, the success of computer systems. Modern computer technology requires professionals of every computing specialty to understand both hardware and software. The interaction between hardware and software at a variety of levels also offers a framework for understanding the fundamentals of computing. The performance of future software systems will be dramatically affected by how well software designers understand the basic hardware techniques at work in a system. Thus, compiler writers, operating system designers, database programmers, and most other software engineers need a firm grounding in the principles presented in this course.

IV. Requirements Fulfilled

This is a required course for all computer engineering majors. This course should be taken in the first year.

V. Required Prerequisites

GEIT 1412: Computer Science II

VI. Learning Outcomes

In this course, students learn:

CLO1. Apply computer arithmetic and convert between number systems

CLO2. Analyse the basics of combinational logic circuits and Boolean Algebra
CLO3. Analyse the basics of sequential logic circuits and their applications
CLO4. Apply basic assembly language constructs
CLO5. Explain and illustrate the top level views of computer functions, interconnections and IO Subsystems
CLO6. Appraise the design aspects of computer memory
CLO7. Develop the communication, leadership and teamwork skills necessary to work in, or in charge of teams

VII. Assessment Strategy

This course is designed with three primary goals in mind: to introduce students to organization and structure of computer systems, to provide students with significant experience in low level (assembly language) control of a computer system, and to provide students with the opportunity to communicate their understanding to their peers in a classroom setting. With this in mind, the course grade involves an assessment of their performance on exams that focus on hardware structures, formal descriptions of systems and low level programming capabilities.

- Weekly assigned homework to motivate students to do the work and earn credit accordingly.
- Weekly, in-class presentations by students related to independent literature research on aspects of the course material and classroom discussion and critique of the presentation.
- Two in-class examinations to assess the student's accumulative mastery of content covered prior to the time of the examination.
- Three major programming assignments to test the student's understanding of the major concepts introduced during the course. Each programming assignment is assessed through instructor and peer review during in-class presentations.
- A comprehensive final examination to assess the student's accumulative mastery of course material.

The final grade is based on 10% credit for the homework, 30% on in-class examinations, 15% on Quizzes and 25% for the final examination. Project work is allotted 20%.

Students are required to maintain a journal of thoughts and commentaries during the course. The journal contains daily entries including the identification of areas of interest and concern, notes on the preparation of presentation and comments and analysis of classmate's presentations. The journal is reviewed weekly by the instructor to provide feedback to the students.
Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he proceeds through the curriculum.

Assessment:

Assignments	10%
Quizzes	15%
Major-1	15%
Major-2	15%
Final exam	25%
Project	20%

Grading Scale: Standard

A+	96 - 100%
A	90 - 95 %
B+	86 – 89 %
B	80 – 85 %
C+	76 – 79 %
C	70 – 75 %
D+	66 – 69 %
D	60 – 65 %
F	Below 60 %

Class Rules:

- Disruptive Behavior
  - Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
  - Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.
  - Eating / Drinking / Smoking: students are requested to refrain from engaging in these activities while in class.

- Class attendance
  - Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

- Tardiness
  - If a student is late for 3 times it is counted as one absence.

- Make-up
  - Midterm & Final exams – different from and harder than in-class exams. You should a valid excuse as per PMU policies

- Late submissions – score reduced by 10% points per day after due date. No work will be accepted after one week of the due date.

- Withdrawal
  - It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

- Academic Honesty and Integrity
  - The instructor strictly adheres to all university policies regarding academic integrity.
  - Academic dishonesty includes but not limited to the following:
    - Cheating on examination or other academic work,
    - Plagiarism and
    - Collusion which means unauthorized collaboration with another in preparing

work offered for academic credit.

- Academic dishonesty will not be tolerated and the PMU academic regulations will be strictly applied.
- Unless specifically expressed by the instructor, collaboration between students in this course, between students in previous courses, external assistance in any form or presenting resources/research without proper citation which has been developed by another individual or organization is strictly prohibited. ALL WORK MUST BE THE RESULT OF YOUR OWN EFFORTS.

➢ Miscellaneous

Mobile phones: A student whose mobile rings during class will be asked to leave the classroom and will receive ½ an absence. Should this happen during an exam, the student will not be allowed to retake the exam at another time, while at the same time receiving a full absence.

VIII. Course Format

This course is primarily a lecture course. Students are expected to attend three hours of lecture per week.

Classroom Hours (3 hours per week) Class: 3
Lab: 0

Web supplement: Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:

- Course syllabus.
- Course assignments.
- Sample solutions to examinations (after graded and returned).
- Course calendar (an active utility).
- Course e-mail (an active utility).
- Course discussion list (an active utility).
- Student course performance (an active utility).

IX. Topics to be Covered

A. Structure and functions of computer
B. History, design and performances of computers
C. Computer component, functions, Registers, instruction
D. Interrupt , Computer Modules, Buses, computer memory
E. Cache memory principles, Element of cache
F. RAM, ROM
G. External memory and I/O
H. Operating System
I. Computer arithmetic
J. Instruction sets

X. Laboratory Exercises

There are no lab exercises for this course.

XI. Technology Component

This course makes use of the university's wireless access infrastructure. The course relies on the university and the students having access to professional grade application development environments

for the students to use. The course's laboratory component requires dedicated signal processing hardware and tools within one of the university provided computer laboratories.

XII. Special Projects/Activities

Students are required to do a research project on one of the main areas related to computers and present it in the class.

XIII. Textbooks and Teaching Aids

A. Required Textbook

William Stallings, (8th Edition). Computer Organization and Architecture: Designing and Performance, ISBN 0131856448

B. Alternative Textbooks

a. Miles J. Murdocca and Vincent P. Heuring, Principles of Computer Architecture, Prentice-Hall, 2000, ISBN 0201436647

b. Richard C. Detmer, Introduction to 80x86 Assembly Language and Computer Architecture, Jones and Bartlett, 2001 ISBN 0763717738

C. Supplemental Print Materials

None

D. Supplemental Online Materials

As available from publisher.

GEIT 3351: Principles of Software Engineering

Semester Credit Hours: 3 (3, 0)

I. Course Overview

The course is designed to provide an introduction to the theory and practice of software development and maintenance. The focus is on the full software development life cycle, including coverage of tools, techniques, principles, and guidelines for software requirements, specification, design and implementation. Particular emphasis is placed on the principles and methods used to develop and validate software requirements. Students are guided toward a better understanding of the various tasks and specialties that contribute to the development of a software product.

II. PMU Competencies and Learning Outcomes

This course helps students develop the ability to become conversant with software engineering topics and acquire a strong understanding of the software development life cycle. In the process, students gain an appreciation for the issues that are important for software engineering practitioners around the world. Software development is largely a collaborative effort, often times spanning multiple countries. It also requires logical, analytical and creative thinking skills to design robust software as well as strong technical writing skills to generate proper design documents. Therefore, the course provides opportunities to help strengthen these skills in students. Additionally, the course provides the students with the communication, leadership, and teamwork skills necessary to effectively work as professionals in teams responsible for software engineering projects.

III. Detailed Course Description

The course begins with a discussion of ethical issues, legal issues, and aspects conducive to effective teamwork, in the context of software engineering projects, with a primary focus on design and implementation issues. It then proceeds with an overview of software engineering process and looks at the software life cycle from an object-oriented perspective. Several life cycle models are discussed, including the Waterfall model, prototyping, spiral model and RAD. Students are introduced to object-oriented modeling and requirements analysis leading to the preparation of requirements specifications. Next, the course introduces students to the software design process, covering topics in process architecture, UI and class design, and culminating into design specifications. The course concludes with a discussion of an implementation plan. Students are introduced to various implementation approaches, concepts of testing and developing a test plan. A brief introduction to project management topics is given to illustrate the final phase of the software development life cycle. GEIT 4351, Software Engineering II, picks up from here and goes into the details project management and actual implementation.

IV. Requirements Fulfilled

This course is required of all students pursuing degrees in computer engineering, computer science, and information technology.

V. Required Prerequisites

- GEIT 1411: Computer Science I
- GEIT 1412: Computer Science II.

VI. Learning Outcomes

In this course, students learn:

1. To describe software engineering topics, and explain the related terms and issues relevant to software engineers around the world.

2. To explain and demonstrate techniques for eliciting and writing requirements specifications; verifying and validating requirements.
3. To develop a robust and scalable solution that satisfies the client requirements; encompassing its design, implementation and testing
4. To use the appropriate software development life cycle and develop a comprehensive project plan and schedule.
5. To develop communication and teamwork skills necessary to accomplish a software project.

VII. Assessment Strategy

The student's performance in this course may be assessed on the basis of:

- One midterm and one final examination (50%).
- Two or three quizzes (10%)
- One project completed outside of class (20%) and evaluation report (5%).
- Homework (10%)
- Class participation (5%).

Relative weights assigned to these items in determining student's final grade are suggested as follows:

- Each exam accounts for 25% of the grade. Combined, the two examinations account for 50% of the grade. Quizzes account for 10% of the grade
- The team project accounts for 20% of the grade. Evaluation report accounts for 5%
- Class participation and homework accounts for 15% of the final grade.

The examinations are designed to assess the mastery the concepts and methods discussed in class, which are primarily based on textbook material. This is complemented by the class discussions on recent articles taken from online industry publications, which allow the students to become conversant with the industry-specific issues related to software engineering. The final team project provides an experience where concepts, methods, and industry-relevant issues are all brought together in a much applied manner to design a software product. Using one of the software design methodology (for example Waterfall life cycle model), students develop all life cycle deliverables, requirement documents, specification and design documents. This set of documents comes at the starting point for the project in GEIT 4351: Software Engineering II, where students design, prototype, test and implement the system.

VIII. Course Format

This course utilizes a mix of lectures and class-discussions to help students learn the various tasks involved in the software development process. The lectures cover several topics outlined later in this syllabus. The class discussions are based on recent articles taken from online industry publications, which are freely available from the Web. The instructor provides the links to the articles, which are then downloaded by the students and read prior to the class.

In addition, the instructor should consider creating a Web site for this course using Web technologies such as BLACKBOARD. At minimum, the site should include:

- Course syllabus.
- Lecture material (for example PowerPoint slides, lecture notes, etc.). These should be placed on the site ahead of class meeting so that students may use the material to prepare for the lecture.
- Course calendar.
- Mechanism to communicate electronically (for example e-mail)
- Discussion groups.
- Project-related Resources
- Students course performance measures.

Classroom Hours (3 hours per week) Class: 3

IX. Topics to be Covered

Week	Chapters	Topics
1	Introduction/ Chapter 1	Introduction to Software Engineering/ Ethical Issues in Software Engineering
2	Chapter2	Software Processes
3	Chapter 3	Agile Software Development
4-5	Chapter 4	Requirements Engineering
6-7	Chapter 5	System Modelling using UML
8	Chapter 6	Architectural Design **(Major 1)**
9-10	Chapter 7	Design and Implementation
11-12	Chapter 8/ Chapter 9	Software Testing/ Software Evolution
13-14	Chapter 23/ Chapter 25	Software Project and Configuration Management
15		Revision, Project Presentation
16	Final Exam	**(Final exam. Scheduled by the Registrar)**

X. Laboratory Exercises

There is no laboratory component in this course.

XI. Technology Component

A. In class, the instructor makes use of state-of-the art multimedia projection equipment and software. These are used to project slides and Web-based content relevant to Web server administration.

B. Outside class, the instructor uses Web-based course management software (for example WebCT, BLACKBOARD) to interact with students as described under course format section.

XII. Special Projects/Activities

Software Engineering is all about developing quality, robust, and secure software. A semester project is intended to give students a feeling of how software is really being developed. In addition, the ability to work with other software developers is essential and critical in a software development project. Therefore, students are required to work effectively in teams throughout the semester and prepare the following deliverables for a proposed system in two phases:

- In Phase one (about eight weeks), students prepare:
  - Requirements Specifications,
  - Design specifications
  - Implementation plan
  - A prototype of the software
- In Phase two (next eight weeks), the complete sets of documentation are redistributed amongst the teams. So each team gets the requirements, design and software written by another team. Each team is now asked to evaluate the set of documents, test the software, and write an evaluation report.
- At the end of the project, each team gives a presentation of its findings.

- Some sample project ideas are as follows:
  - Transportation Management System: Develop a software based system to manage assignments of drivers to buses, scheduling, a mechanism to identify valid customer, and notify customers for any change in bus schedule etc.
  - Vehicle Rental Management System: Develop a software based system to manage vehicle rental search, reservation and notification based system
  - Library Management System: Develop a system to manage library resources and check-in/check-out mechanism of these resources
  - Course Management System: Develop a system to manage courses, add course material, assign grades, communicate with students etc.
  - Housing Management System: Develop a system to manage a housing compound with unit allocation, maintenance and rent system.

XIII. Textbooks and Teaching Aids

A. Required Textbook

Ian Sommerville; Software Engineering, 10/E; Pearson , 10th edition, April 2015

ISBN-10: 0133943038 • ISBN-13: 9780133943030

B. Alternative Textbooks

1. Craig Larman; Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition); Prentice Hall

   ISBN-10: 0131489062, ISBN-13: 978-0131489066

C. Supplemental Print Materials

Martin Fowler; UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition); Addison-Wesley Professional; ISBN-10: 0321193687, ISBN-13: 978-0321193681

D. Supplemental Online Materials

Recent articles taken from online industry publications such as the Methods and Tools, and CTO magazines. The instructor provides the links to the articles, which are freely available from the Web.

GEIT 4361: Practical Training

Semester Credit Hours: 3 (3,0)

I. Course Overview

This course provides opportunities for students to apply the academic concepts, skills and techniques learned in their coursework to a professional work-oriented setting. The course offers the potential for a one-semester internship with a regional employer or a directed study course providing practical learning experiences that benefit the community.

II. PMU Competencies and Learning Outcomes

GEIT 4361: Practical Training is primarily practical in nature with no lecture content. Students' experience will be based on the individual placement of the student into an internship or directed practical training. This course requires students to apply all the PMU competencies (including professional competence, critical thinking, communication, leadership, and teamwork) to an organization. The internship is designed to sharpen the student's analytical and problem solving skills through research and guidance from the instructor and the business to which the student is assigned. This course allows for integrating different subject areas to deal with a problem or a situation that is not one-dimensional.

III. Detailed Course Description

The role of the practical training is to provide students with an appreciation of the types of work involved with their major before they actually enter the job market. The practical training course also provides students with first-hand experience and supplements the theories they have learned in the classroom. It allows them to draw upon various concepts to solve complex, real world problems. It provides the business or consumer with an opportunity to have students with fresh ideas work on an issue or a problem currently facing them. Internships may also provide employers with a risk-free chance to try potential employees before actually hiring them.

IV. Requirements Fulfilled

GEIT 4361: Practical Training satisfies three hours of the requirements for degrees in Information Technology, Computer Science, and Computer Engineering. It is an elective course available to all students pursuing a degree programs within the College of Computer Engineering and Science. This course should be taken in the senior year.

V. Required Prerequisites

Senior year standing and the consent of the instructor.

VI. Learning Outcomes (modified Oct 2015 as per ABET)

Conduct professional oral presentation in chosen field area. (Presentation Rubric) 30
Write a professional technical report. (Writing rubric) 20 + (Employer) 5
Identify, learn and apply contemporary computing solutions for business problem (Presentation Rubric) 10 + (Writing rubric) 10 + (Employer) 5
Recognize the importance of ethics and professional conduct in a business (Logbook) 10 + (Employer) 5
Apply communication, collaboration and teamwork skills necessary for effectively working in a professional environment. (Employer Rubric) 5

VII. Assessment Strategy

Internship
A faculty member supervises the student and monitors his or her progress. The employer provides written feedback on the student's performance and professional competencies and skills emphasized by the PMU. This feedback enables the instructor to assess student work and assign a grade for the course.
Directed Study
If it not possible for the student to participate in an internship, the student may be allowed to take a directed study course. Such a course will be designed to provide practical learning experience under the supervision of a faculty member and targeted to designing solutions that meet the real world technology needs of the local community.
To facilitate directed studies opportunities, is recommended that the PMU establish a Community Technology Resource Center to encourage members of the community to seek professional advice for small technology projects that the students can complete under the supervision of the faculty.

VIII. Course Format

Internship
The student works specified hours at an employer's location. Total hours devoted to this activity should be 10 hours a week for one semester. The workload includes reports and presentations.

Students are expected to file formal reports and make presentations to the supervising faculty each month.

Directed Study
Directed study students are expected to develop a schedule equivalent to 150 total hours in managing their project. The schedule is to be validated and approved by the supervising faculty.

Students are expected to file formal reports of their progress and make presentations to the supervising faculty each month.

Internship Hours (10 hours per week)
Directed Study (10 hours per week)

IX. Topics to be Covered

No new topics are addressed. Rather, the course involves application of course material learned in the College of Computer Engineering and Science core curriculum and the student's major field of specialization.

X. Laboratory Exercises

This course does not require a separate laboratory. However, Internship students should have access to the technology infrastructure at the business location and directed study students should have access to the university technology infrastructure.

XI. Technology Component

Technology is used as needed and depends on the level of technology present at the business location.

XII. Special Projects / Activities

Students are required to keep a "reflective notebook" in which, after each session with the employer, client or supervising faculty, they enter their own assessments of what they learned, and what questions remain. Students should use their notebooks as tools for identifying problems and evaluating potential solutions

XIII. Textbooks and Teaching Aids

A. Required Textbook
None.

B. Alternative Textbooks
None.

C. Supplemental Print Materials
None.

D. Supplemental Online Materials

None

ASSE 4311: Learning Assessment III

Semester Credit Hours: 3 (3,0)

I. Course Overview

This is the capstone course required of all students pursuing an undergraduate degree program within the College of Computer Engineering and Science. The objective of this course is to bring together in an applied manner the knowledge and skills obtained by the students throughout their undergraduate program. The course is designed so as to cover topics that are relevant from an integrated IT systems design and implementation perspective. The term "integrated IT systems design and implementation" refers to complex collaborative efforts that bring together knowledge skills in the related areas of computer science, computer engineering, and information technology (as covered by the three undergraduate programs offered by the College of Computer Engineering and Science). The course is very applied. One of its main components is a team project focusing on integrated IT systems design and implementation. The course also includes a mix of speakers' presentations, project work, and discussions on contemporary articles from industry publications.

II. PMU Competencies and Learning Outcomes

This course helps students develop the ability to become conversant with integrated IT systems design and implementation topics. Additionally, the course provides the students with the communication, leadership and teamwork skills necessary to effectively work as professionals in teams, or in charge of teams, responsible for integrated IT systems design and implementation projects. Finally, the course looks at IT from a strategic and integrated perspective, that is, a holistic or whole-organization perspective, rather than viewing IT as a set of tools used in a localized manner to improve certain parts of an organization (for example, a specific department, area, or process).

III. Detailed Course Description

The course comprises a set of speakers' presentations and class discussions on contemporary articles from industry publications addressing integrated IT systems design and implementation. Particular attention is paid to the complex nature of integrated IT systems design and implementation projects. Such projects usually involve a high level of heterogeneity in terms of technologies used, cultural background of those participating in the project, and types of expertise involved. This course emphasizes a holistic or whole-organization perspective, rather than the more common perspective of IT as a set of tools to be used in a localized manner to improve parts of an organization that do not usually interact

with each other (for example, a specific department, area, or process). The goal with this emphasis is to provide students with a realistic view of what the job of a chief technology officer (CTO) entails. CTOs are usually responsible for managing the IT resources of an entire organization, and ensuring that those resources are used in a synergistic way to benefit the organization as a whole.

IV. Requirements Fulfilled

This course is required of all students pursuing degrees in computer engineering, computer science, and information technology.

V. Required Prerequisites

- GEIT 1411: Computer Science I
- GEIT 1412: Computer Science II
- GEIT 3331: Computer Organization I
- GEIT2291: Professional Ethics
- GEIT 3341: Database I
- GEIT 3351: Principles of Software Engineering

VI. Learning Outcomes

In this course, students learn:

- Analyze a business problem, identify the user needs and define the appropriate IT solution.
- Design, implement, and integrate a computer-based system to meet requirements.
- Demonstrate communication, leadership and teamwork skills necessary for effectively working as professionals in teams, or in charge of teams.
- Recognize the importance of ethics and professional conduct.
- Develop an effective project plan.
- Design and deliver a quality presentation.
- Write a technical document to a professional level.

VII. Assessment Strategy

Students are assessed based on: (a) their class participation, which includes active participation in speakers' presentations, and the discussion of recent articles taken from online industry publications; and (b) the quality of a final team project and related oral presentation. The relative weights of each of these items on the final grade are as follows:

- Class participation accounts for 30% of the grade, and is evaluated based on the student's active participation in speakers' presentations, and the ability of students to add to the material already provided by the instructor to them.
-

- The final team project accounts for 70% of the grade. It is evaluated based on a project document, oral presentation, and client perceptions of the team project. The project must be conducted in collaboration with a client organization (for example, a department at a large company or non-profit organization). A letter from the main contact person at the client organization, discussing and evaluating the project and its outcomes, must be provided to the instructor. The letter should contain the contact information of the person writing so the instructor can call him/her up and inquire about the project.

The speakers' presentations and the discussion of recent articles taken from online industry publications allow the students to become conversant with the industry-specific terminology related to integrated IT systems design and implementation issues. The final project provides an experience where concepts, methods, and industry-relevant issues are all brought together in a very applied manner to solve a real problem faced by a real organization. It gives the students the necessary exposure to industry-relevant issues to prepare them for the future challenge of pursuing a successful career as IT professionals.

VIII. Course Format

In 50% of the class meetings, invited speakers with practical industry experience (ideally senior managers) give presentations on topics related to integrated IT systems design and implementation. These presentations last approximately 45 minutes, followed by 15 minutes for questions and answers. The remainder of the time in class is used for project work and/or class discussions. The remaining class meetings are used for project work and/or class discussions. The latter are based on recent articles taken from online industry publications such as the CTO and CIO magazines, which are freely available from the Web. The instructor provides the links to the articles, which are then downloaded by the students and read prior to the class. In class, the students discuss the articles in small teams for about 20 minutes, developing three provocative questions per team. This is followed by a discussion involving the whole class, where each team asks one of the questions they developed, and other teams answer them, until all teams asked at least one of their questions. This discussion format is likely to lead to lively debate on topics that are directly addressed by the article, as well as on topics that are indirectly related to the article.

Classroom Hours (3 hours per week) Class: 3
Lab: 0

IX. Topics to be covered

A. Ethical issues, legal issues, and effective teamwork
   1. Ethical and legal issues in integrated IT design and implementation
   2. Typical integrated IT design and implementation teams
   3. Conflict resolution in integrated IT design and implementation teams
   4. Effective teamwork in integrated IT design and implementation teams

B. Integrated IT design and implementation
   1. Classic cases of integrated IT design and implementation
   2. Contemporary cases of integrated IT design and implementation
   3. Integrated IT design and implementation project management
   4. Best practices in integrated IT design and implementation
   5. Classic cases of IT use as a strategic weapon
   6. Developing an organization-wide IT strategy

X. Laboratory Exercises

There is no laboratory component in this course.

XI. Technology Component

A. In class, the instructor makes use of state-of-the art multimedia projection equipment and software. These are used to project slides and Web-based content, as well as play freely available Web-based video clips from Web sites covering topics relevant to the class (for example, CNN.com Technology).

B. Outside class, the instructor uses Web-based course management software to interact with students, provide feedback on their performance, make available links to online articles, as well as receive documents (for example, draft versions of project reports) and provide feedback on them.

XII. Special Projects/Activities

The team project consists of meeting with members of a client organization (for example, a department at a large company or non-profit organization), gathering relevant information from them, and conducting an integrated IT systems design and implementation project. The team project also involves the development of a prototype integrated IT solution, and the preparation of a document containing the following elements:

- A set of organizational problems that could potentially be solved through an integrated IT systems design and implementation project. This project must bring together topics that are unique to each of the sister areas of computer science, computer engineering, and IT applications (covered by the three undergraduate programs offered by the College of IT). It is strongly recommended that student teams include individuals representing each of the three undergraduate majors of the College of IT.
- A detailed specification of an integrated IT solution to the problems above. This specification should include hardware and software details, as well as details in connection with how the solution is integrated with existing technologies and systems in the client organization.
- A detailed description of the costs and potential benefits, from an organizational perspective, associated with the integrated IT solution.
- A detailed plan for the implementation of the integrated IT solution, including a project task decomposition and related budgeting, a project timeline and related chart, and project risk management plan (with contingency options, in case the project is not successfully implemented as planned).

The integrated IT solution actually implemented (in a prototyped manner) should incorporate about

60-80% of the functionality of the solution proposed by the student teams in their reports.

Oral presentation. Teams summarize and explain the information contained in their project document in an oral presentation in class at the end of the semester. This oral presentation should also incorporate a demonstration of the integrated IT solution actually implemented by the students.

XIII. Textbooks and Teaching Aids

A. Required Textbook

Tom Demarco and Timothy Lister, Peopleware: Productive Projects and Teams; Dorset House; 2nd edition (February 1, 1999)
ISBN: 0932633439.

B. Alternative Textbooks

John Ward and Joe Peppard, Strategic Planning for Information Systems; John Wiley & Sons; 3rd edition (June 15, 2002)
ISBN: 0470841478.

C. Supplemental Print Materials

Roger Beach, Adopting Internet technology in manufacturing: a strategic perspective, Production Planning & Control, Jan 2004, Vol. 15 Issue 1, p80, 10p.

D. Supplemental Online Materials

Recent articles taken from online industry publications such as the CTO and CIO magazines. The instructor provides the links to the articles, which are freely available from the Web.

C.

D. INFORMATION TECHNOLOGY COURSES

- ITAP 1311: Introduction to IT
- ITAP 2312: Web Programming
- ITAP 2431: Network Management
- ITAP 3431: Network Security
- ITAP 3313: User Interface Development
- ITAP 3471: Web Server Administration
- ITAP 3411: Systems Programming
- ITAP 3382: Business Intelligence
- ITAP 3383: Enterprise Resource Planning Systems
- ITAP 4371: E-Commerce

IT Elective courses being offered:

- ITAP 4316: Introduction to Software Project Management
- ITAP 3371: Database II
- ITAP 4313: Advanced Visual Programming
- ITAP 3362: Mobile Application Design & Development
- ITAP 4315: Cloud Computing
- ITAP 4361: Operating Systems
- ITAP4393: Special Topics

- ITAP 3315: Software Testing & Quality Assurance
- ITAP 4391: Reuse and Component-based Development (Never offered)
- ITAP 3363: Principles of Open Source & Enterprise Computing (Never offered)
- ITAP 4314: Multimedia Computing and Applications (Never offered)
- ITAP 4390: Information Security and Assurance (Never offered)
- ITAP 4396: Computer and Network Forensics (Never offered)
- ITAP 4376: Secure E-commerce (Never offered)
- ITAP 4362: Network Administration (Never offered)
- ITAP 4363: Systems Administration (Never offered)
- ITAP 4367: Database Administration (Never offered)

ITAP 4364: Performance Analysis of Computer Networks (Never offered)
ITAP 1311- Introduction to IT

Prerequisites

None

Credit hours

3 Credit Hours

Course overview

This course is designed to provide the students with an introduction to information systems. It includes a discussion of the fundamental principles, generalizations, and theories of Information Technology. Students are exposed to many of the areas of information Technology including databases, telecommunications and networking, software and hardware concepts, and information security.

PMU Competencies and Learning outcomes

By the end of the course the students should be able to:

CLO 1: Define concepts and terminology related to information technology

CLO2: Appraise the role of Information Technology and its applications in a business environment

CLO3: List and contrast various types of information technology concepts such as telecommunications, Database, Networking and software development

CLO4: List, evaluate and select information resources and technological solutions based on their appropriateness to specific tasks

CLO5: Use a variety of media and formats, including telecommunications, to collaborate, publish, and interact with peers, experts, and other audience
CLO6: Use technology tools to process data and report results

Detailed course description

This course is designed to provide the students with an introduction to information systems. It includes a discussion of the fundamental principles, generalizations, and theories of Information Technology. Students are exposed to many of the areas of information Technology including databases, telecommunications, Networking, software and hardware concepts, and information security.

Required Text

Gary B. Shelly and Misty E. Vermaat, Discovering Computers 2014: Living in a Digital World, Course Technology, Cengage Learning, 2010. ISBN 13: 978-0-495-80681-3, ISBN 10: 0-495-80681-1

Class RULES

Disruptive Behavior

Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance

Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness

When a student is late for 3 times it is counted as one absent.

WITHDRAWAL

It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment

Student's performance in this course may be assessed on the basis of:
Homework
Two term exams
A programming project
Lab exercises
A final exam

Evaluation

A maximum of the final grade according to PMU Grading System, computed as follows:

Participation	5%
Homework	15%
Quiz	15%
Project	20%
Mid-term	20%
Final	25%
Total	100%

A. Instruction

This course utilizes both lecture/discussion and laboratory exercises. Students are expected to attend three hours of lecture/discussion per week and three hours of laboratory per week. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Occasionally, students have a collaborative problem solving activity.

B. Web supplement

Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:
Course syllabus
Course assignments
Sample solutions to examinations (after being graded and returned)
Sample solutions to programming exercises (after being graded and returned)
Course calendar (an active utility)
Course e-mail (an active utility)
Course discussion list (an active utility)
Student course performance (an active utility)

CLASS SCHEDULE (Tentative)

List of Topics	No. of Weeks
The Internet & World Wide Web	2
Application Software	3
The Components of the System Units	3
Input	1
Output	1
Storage	1
Operating System and Utility Programs	2
Computer and Network	1
Database Management	1

ITAP 2312: Web Programming

Prerequisites
GEIT 1411: Computer Science I

Credit hours
4 Credit Hours

Course overview

This course is designed to provide the students with an introduction to World Wide Web programming. It introduces the student to the techniques used in programming web pages for interactive content. It specifically addresses the basic elements of AJAX -Asynchronous JavaScript and XML to design web pages that dynamically interact with databases that reside on a server. The course begins by reviewing basic web technologies such as HTML, XHTML, CSS style sheets, and explores the use of event-driven programming in JavaScript to add interactive elements such as buttons and text fields to web pages. Students will be able to use AJAX tools to build web pages that connect to servers like Google to dynamically access data (maps, search results, videos, images, etc).

PMU Competencies and Learning outcomes

By the end of the course the students should be able to:
Construct a web page using the latest HTML technology
Design a web page with the suitable formatting technology e.g., CSS3
Apply event-driven programming in web application development
Develop and implement a web application using JavaScript framework (for example JQuery)
Develop the communication, leadership and teamwork skills necessary to work in or lead teams

Detailed Course Description

ITAP 1312: The course begins by reviewing basic web technologies such as HTML, XHTML, CSS style sheets, and explores the use of event-driven programming in JavaScript to add interactive elements such as buttons and text fields to web pages. Students will be able to use AJAX tools to build web pages that connect to servers like Google to dynamically access data (maps, search results, videos, images, etc).

Required Text

A. Required Textbook: Deitel, J. & Deitel, M. (2009). Internet & World Wide Web How To Program; 5tht edition. Pearson Education. ISBN 0-13-603542-6

Class Rules

Disruptive Behavior
Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance
Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness
When a student is late for 3 times it is counted as one absent.

Withdrawal
It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment
Student's performance in this course may be assessed on the basis of:
Homework
Two term exams
A programming project
Lab exercises
A final exam

Evaluation

Homework Assignments	5%
Programming Project	15%
Lab	20%
Exam 1	15%
Exam 2	15%
Final	30%
Total	100%

Course Format

A. Instruction
This course utilizes both lecture/discussion and laboratory exercises. Students are expected to attend three hours of lecture/discussion per week and three hours of laboratory per week. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Occasionally, students have a collaborative problem solving activity.

B. Web supplement
Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:
Course syllabus
Course assignments
Sample solutions to examinations (after being graded and returned)
Sample solutions to programming exercises (after being graded and returned)
Course calendar (an active utility)
Course e-mail (an active utility)
Course discussion list (an active utility)
Student course performance (an active utility)

CLASS SCHEDULE (Tentative)

Week#	Lecture (3 hours/week)	Lab (2hours/week)	Projects/ Readings
Week #1	Syllabus Introduction to Computers and the Internet	Laboratory exercises	Read Chapter 1
Week #2	Web Browser Basics: Internet Explorer and Firefox Dive into® Web 2.0	Laboratory exercises	Read Chapter 2 Read Chapter 3
Week #3	Dive into® Web 2.0 HTML Basics	Laboratory exercises	Read Chapter 3
Week #4	Introduction to XHTML	Laboratory exercises	Read Chapter 4
Week #5	Introduction to XHTML Cascading Style Sheets™ (CSS)	Laboratory exercises	Read Chapter 4 Read Chapter 5
Week #6	Cascading Style Sheets™ (CSS)	Laboratory exercises	Read Chapter 5
Week #7	JavaScript: Introduction to Scripting	Laboratory exercises	Read Chapter 6
Week #8	JavaScript: Introduction to Scripting	Laboratory exercises	Read Chapter 6
Week #9	JavaScript: Control Statements I	Laboratory exercises	Read Chapter 7
Week #10	JavaScript: Control Statements II	Laboratory exercises	Read Chapter 8
Week #11	JavaScript: Functions	Laboratory exercises	Read Chapter 9
Week #12	JavaScript: Arrays	Laboratory exercises	Read Chapter 10
Week #13	JavaScript: Objects	Laboratory exercises	Read Chapter11
Week #14	Document Object Model (DOM): Objects and Collections	Laboratory exercises	Read Chapter 12 Group Project presentations

Week #15	JavaScript: Events	Laboratory exercises	Read Chapter 13 Group Project presentations

ITAP 2431: Network Management

Prerequisites

- GEIT 1412: Computer Science II

Credit hours

4 Credit Hours

Course overview

This course introduces students to business data communications and networking concepts, tools and methods. The goal is to prepare students to apply networking tools and methods to the solution of business problems. The course covers the use of basic data communications and networking hardware such as hubs and routers, and of simple programming tools to customize and integrate existing software. It introduces networking, distributed transaction processing, and Web-related concepts. The course also covers concepts for managing distributed storage and connectivity related to data, voice, image, and video. Its specific focus is on Web-based systems. The course includes a mix of lectures, laboratory demonstrations and assignments, and discussions on contemporary articles from industry publications.

PMU Competencies and Learning outcomes

This course helps students become conversant with network management issues and understand the related terms and issues that are important for network management professionals around the world. Additionally, the course provides the students with the communication, leadership and teamwork skills necessary to effectively work as professionals in teams, or in charge of teams, responsible for operating complex network environments. Finally, the course imparts on the students an understanding of networks as more than webs of switches and cables – namely as resources that support the core and mission-critical business processes of an organization.

Detailed Course Description

The course begins with a discussion of ethical issues, legal issues, and aspects conducive to effective teamwork, in the context of network management. It then proceeds with a review of key introductory topics such as packet switching, types of communication signals, types of networking media and topologies, the OSI Model, and networking protocols. Next the course covers core technological ideas and elements used for networking computers in organizations, such as local area networks (LANs), wireless LANs, wide area networks, hubs, wireless access points, and routers. The course concludes with a discussion of advanced issues in connection multimedia delivery and distributed computing using the Web. The emphasis on this course is on networking infrastructure set up and management, rather than on distributed programming and other programming-related topics.

Learning Outcomes

In this course, students learn:

CLO1: Define and explain reference models and various network protocols.

CLO2: Describe and differentiate between various network technologies including LANs, WLAN, WAN, Backbone, etc.

CLO3: Explain the fundamental principles of data transmission, including transmission media, signal encoding and modulation, error detection and correction, multiplexing

CLO4: Configure, manage, secure, and maintain network infrastructures

CLO5: Develop the communication, leadership and teamwork skills necessary to work in or lead teams

Required Text

A. Required Textbook
Jerry FitzGerald and Alan Dennis, Business Data Communications and Networking; 8th edition (July 9, 2004) John Wiley & Sons
ISBN: 0471348074.

B. Alternative Textbooks
Jerry FitzGerald, Alan Dennis, Alexandra Durcikova; Business Data Communications and Networking, Wiley; 12th Edition (August 15, 2014), ISBN: 978-1-118-89168-1.

Class Rules

Disruptive Behavior
Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance
Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness
When a student is late for 3 times it is counted as one absent.

Withdrawal
It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment
Students are assessed based on: their performance in two exams (midterm and final); their class participation, which includes programming assignment, discussion of recent articles taken from online industry publications; and the quality of a final team project and related oral presentation. The relative weights of each of these items on the final grade are as follows:

GRADING SCHEME:

Participation and preparation	10%
Laboratory Assignments	20%
Quizzes	10%
Final project	20%
Midterm exam	15%
Final Exam	25%
Total	100%

Course Format

A. Instruction

Four of the course's class meetings are used for laboratory demonstrations and activities geared at helping the students learn the several steps involved in designing and implementing a database system. The other class meetings are split into two main components: lectures, and class discussions. The lectures cover several topics outlined later in this syllabus. The class discussions are based on recent articles taken from online industry publications such as the Searchers and CIO magazines, which are freely available from the Web. The instructor provides the links to the articles, which are then downloaded by the students and read prior to class. In class, the students discuss the articles in small teams for about 20 minutes, developing three provocative questions per team. This is followed by a discussion involving the whole class,

where each team asks one of the questions they developed, and other teams answer them, until all teams asked at least one of their questions. This discussion format is likely to lead to lively debate on topics that are directly addressed by the article, as well as on topics that are indirectly related to the article.

B. Web supplement

Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:
Course syllabus
Course assignments
Sample solutions to examinations (after being graded and returned)
Sample solutions to programming assignments (after being graded and returned)
Course calendar (an active utility)
Course e-mail (an active utility)
Course discussion list (an active utility)
Student course performance (an active utility)

CLASS SCHEDULE (Tentative)

Week#	Lecture (3 hours/week)	Lab (3hours/week)
Week #1	Syllabus INTRODUCTION Introduction to Data Communications	Laboratory exercises
Week #2	Application Layer	Laboratory exercises
Week #3	Application Layer	Laboratory exercises
Week #4	Physical & Data Link Layers- Part 1	Laboratory exercises
Week #5	Network and Transport Layers	Laboratory exercises
Week #6	Network and Transport Layers	Laboratory exercises
Week #7	Local Area Networks	Laboratory exercises
Week #8	Local Area Networks	Laboratory exercises
Week #9	Wireless Area Networks	Laboratory exercises
Week #10	Physical Layer & Data Link Layer- Part 2	Laboratory exercises
Week #11	Backbones	Laboratory exercises
Week #12	Wide Area Networks	Laboratory exercises
Week #13	Wide Area Networks	Laboratory exercises
Week #14	The Internet	Laboratory exercises
Week #15	Network Management	Laboratory exercises

ITAP 3431: Network Security

Semester Credit Hours: 4 (3, 3)

I. Course Overview

This course examines the basic principles, techniques and technologies associated with securing local area networks. Topics covered include security threats, data protection including cryptography and authentication, a review of network security applications and techniques for the management of intruders, malicious software and other internal and external threats to the network

II. PMU Competencies and Learning Outcomes

This course is highly practical in nature. Effective management of network security is a matter of practical skill and effective management and oversight as much as it is academic. This course primarily addresses two of the PMU competencies, critical thinking and teamwork. Network security requires the examination of network activity and response to potential threats as they arise. Team-based responses are essential to ensure that security is not dependent upon a single individual and so the course models the professional environment.

III. Detailed Course Description

ITAP 3431: Network Security examines the techniques used by network administrators and network security technicians in protecting and securing local area networks. Techniques for ensuring the confidentiality of messages are examined, including public and private key encryption, message and target authentication, digital signatures and key management issues. Network security applications are discussed in detail including authentication, email and web security, IP security and network management security. Techniques for detecting and defending against intruders, malicious software and other forms of attack are also covered.

IV. Requirements Fulfilled

This course satisfies three hours of the requirements for degrees in Information Technology. It is an available elective for the degree in Computer Science and Computer Engineering. It should be taken no earlier than the junior year.

V. Required Prerequisites

ITAP 2431 Network Management

VI. Learning Outcomes

In this course, students learn:

- To describe available cryptographic and authentication techniques for securing network communications
- To explain and differentiate various security attacks (such as DoS, spoofing, hijacking, etc) and malware (such as viruses, worms, Trojan horses, zombies, logic bombs, etc).
- To apply cryptographic and authentication techniques in solving practical problems.
- To analyze, propose and justify solutions for new and unexpected security threats.
- Develop the communication, leadership and teamwork skills necessary to work in teams, or in charge of teams

VII. Assessment Strategy

This course is designed to introduce students to the concepts and practical skills and techniques involved in securing local area networks from internal and external threats. With this in mind, the course grade involves an assessment of their performance on, and understanding of the threats, the application of techniques for minimizing risk associated with those threats and to the solution of security problems and the communication of designed solutions to those problems to an audience. Course grades are be based on

- Weekly assigned homework to motivate students to do the work and earn credit accordingly.
- Weekly structured laboratory exercises designed to guide students through specific course topics.
- Several in-class quizzes/exams to assess students' accumulative mastery of content covered prior to time of exam.
- A comprehensive final exam to assess students' accumulative mastery of course material.

Students' final grades are based on 10% credit for homework and project, 5% for participation in classroom discussion, 20% for weekly lab exercises, 25% on major exams, 15% on quizzes and 25% for the final examination.

Students are required to maintain a journal of thoughts and commentaries during the course. The journal contains daily entries including the identification of areas of interest and concern, notes on the preparation of presentation and comments and analysis of classmate's presentations. The journal is reviewed weekly by the instructor to provide feedback to the students.

Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he proceeds through the curriculum.

VIII. Course Format

This course utilizes both lecture/discussion and laboratory exercises. Students are expected to attend two hours of lecture/discussion per week and three hours of laboratory per week. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Once a week students should have at least 30 minutes of collaborative problem solving activity.

Classroom Hours (6 hours per week) Class: 3
Lab: 3

Web supplement: Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:
Course syllabus
Course assignments
Sample solutions to examinations (after being graded and returned)
Sample solutions to programming assignments (after being graded and returned)
Course calendar (an active utility)
Course e-mail (an active utility)
Course discussion list (an active utility)
Student course performance (an active utility)

IX. Topics to be covered

A. Security Principles and Standards
1. Basic Security Concepts
2. OSI Security architecture
3. ANSI Security Standards

B. Cryptography
1. Symmetric encryption and message confidentiality
2. Public Key Cryptography
3. Authentication

C. Network Security applications
1. Authentication
2. Email and Web Security
3. IP Security
4. Network Management Security

D. System Security
1. Intruder Management
2. Malicious Software
3. Firewall Management

X. Laboratory Exercises

This course requires a weekly 3-hour lab component. Topics to be covered in the laboratory sessions should include:
Security overview – Using NTFS to secure local resources
Device & System Management – Installing service packs and hotfixes. Protecting the systems account database. Configuring Network Settings
Media – transferring NTFS encrypted files, NetMon, autocleaning applications
Authentication – Setting access policies and techniques for bypassing access control
Attacks & malicious code – at, DDOS attacks, Netbus Trojan horses
Remote access – Configuring VPN's, remote access policy
Email – PGP, passphrase caching, public key management and malicious file detection
Web Security – IE security, Content filtering
Directory and file Transfer Services – FTP configuration and restrictions
Wireless and Instant Messaging – wireless security options, telnet management
Network Security topologies – RRAS and NAT, Configuring I/O filters and VLAN's
Intrusion Detection – Detection applications, honeypots
Security Baselines –Defining security templates, IIS lockdown, Security analyzers
Cryptography –certification management
Physical Security – Physical Barriers, biometrics, Social Engineering
Disaster recovery and Business continuity

XI. Technology Component

This course makes use of the university's wireless access infrastructure during the class/lecture sessions. The course relies on the university and the students having access to an isolated professional grade network environment for the students to use.

XII. Special Projects/Activities

Students are required to keep a "reflective notebook" in which, after each class, they enter their own assessments of what they learned, and what questions remain from the class. From each exercise set, each student selects one problem, which the student thinks best reflects the way the mathematical topic is used in a technical context. A detailed solution to the problem is included in the student's reflective notebook.

XIII. Textbooks and Teaching Aids

A. Required Textbook

William Stallings, Network Security Essentials Applications and Standards, 5th Edition, Pearson, ISBN-13: 9780133370430.

B. Alternative Textbooks

None

C. Supplemental Print Materials

PMU Network Security Lab Manual

D. Supplemental Online Materials

As available from publisher.

ITAP 3313: User Interface Development

Prerequisites

GEIT 1411: Computer Science I
GEIT 1412: Computer Science II

Credit hours

3 Credit Hours

Course overview

This is an introductory course to the subject of interaction design. It covers issues like the design of interactive products to support the way people communicate and interact in their everyday and working lives plus the design of spaces for human communication and interaction.
The goal of the course is to expose the students to the various techniques and methods relating to the subject, such that they can participate actively in the production and design of products and spaces that are Usable/User-Friendly. i.e. products and spaces that are both intuitive, easy to learn, effective to use and provide an enjoyable experience
The slides from the 2nd edition of the text book can be found on blackboard.
The course ends with mobile interface design and implementation as an example of use of interaction design activities.

PMU Competencies and Learning outcomes

The goal of this course is to provide the student with a basic knowledge of human-computer interaction as a distinct discipline and to investigate specific issues involving human-computer interaction and user interface design.
The course provides opportunities for technical skill development as well as communication, collaboration and leadership skills through the maintenance of journals, detailing progress in group projects, and in-class presentations
This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students.
The course includes individual as well as group projects and establishes both conceptual reasoning skills and technical communication skills.

Detailed Course Description

Modern computer technology requires professionals of every computing specialty to understand both hardware and software. The interaction between hardware and software at a variety of levels also offers a framework for understanding the fundamentals of computing. The performance of future software systems will be dramatically affected by how well software designers understand the basic hardware techniques at work in a system. Thus, compiler writers, operating system designers, database programmers, and most other software engineers need a firm grounding in the principles presented in this course.
In addition, this course will cover the basic theory and concepts in the area of human-computer interaction.
The course is intended to introduce the student to the basic concepts of human- computer interaction. It will cover the basic theory and methods that exist in the field. Case studies are used throughout the readings to exemplify the methods presented and to lend a context to the issues discussed. The assignments are designed to give the student practice in a set of the more basic techniques used in the human-computer interaction discipline.
The course will begin by introducing the students to the software development process and discussing the behavioral techniques that apply at different stages of this process. It will then discuss the basic applications of these techniques: survey methods, task analyses, usability studies and prototyping. It will finish by discussing a set of innovative interfaces and new developments in human computer interaction. The implementation part of the course reviews the primary concepts of mobile programming. Some background is required in object oriented programming.

Learning Outcomes

After successfully completing the course, students will be able to:

1. List, define and analyze user interface principles and issues and their importance in product design
2. Describe basic techniques for user requirement elicitation, data collection, and analysis
3. Design and implement a user interface according to HCI design principles
4. Evaluate systems design and interfaces from an interaction perspective
5. Develop the communication, leadership and teamwork skills necessary to work in or in charge of teams

Required Text

Required Textbook
Interaction Design: Beyond Human-Computer Interaction; Jennifer Preece, Yvonne Rogers, and Helen Sharp, Wiley and Son, Inc. 2002, ISBN: 0-471- 49278-7.
Alternative Textbooks
Designing the User Interface
4th edition, Ben Shneiderman and Catherine Plaisant
Addison-Wesley, 2005
ISBN: 0-321-19786-0
Human-Computer Interaction
Third Edition by Alan Dix et al
Prentice Hall (2004).
Supplemental Online Materials
http://www.id-book.com/
http://hcibib.org/
http://hcidesigns.com/beck/capstone/

Class Rules

Disruptive Behavior
Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance
Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness
When a student is late for 3 times it is counted as one absent. Students are considered tardy if they arrive after the first 10 minutes (according to the instructor's watch).

Withdrawal
It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W", after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment
There will be individual homework assignments as well as one group project broken up into several phases. Some of the assignments will require programming. There will also be frequent in-class discussions and activities in which students are expected to actively participate. There will be a midterm and a final exam.
The final grade will be calculated this way:
Participation, Assignments and Presentations: 15%
Quizzes: 10%
Major 1: 15%
Major 2: 15%
Project: 20%
Final: 25%

Participation
Attendance does not mean participation, and therefore will not contribute to the participation grade. Participation will be assessed by the instructor on the basis of student's exhibited interest to the course by answering questions in class, asking questions, and actively contributing to classroom discussion.

Grading Scale

A+	96 - 100%
A	90 - 95 %
B+	86 – 89 %
B	80 – 85 %
C+	76 – 79 %
C	70 – 75 %
D+	66 – 69 %
D	60 – 65 %
F	Below 60 %

Make-up
Small homework and in class work – no make-up
Majors & Final exams – different from and harder than the scheduled exams (only if valid excuse is available). Is offered only to students with a serious medical condition.

Late submissions
For each day that the submission is late, it is a 10% penalty.

Course Format

A. Instruction

The course is primarily a lecture-based course in which the students are required to complete significant projects outside of class time. The course will include individual assignments as well as group projects and provide opportunities for the presentation and defense of designed solutions. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Once a week students should have at least 30 minutes of collaborative problem solving activity.

B. Web supplement

Web supplement: Course home page (the university's BLACKBOARD) should contain the following:
Course syllabus.
Course assignments.
Model programmed solutions to programming assignments (once students have completed them)
Course calendar (an active utility).
Course e-mail (an active utility).
Course discussion list (an active utility).
Students course marks. (an active utility).

C. Technology Component

In class, the instructor makes use of state-of-the art multimedia projection equipment and software. These are used to project slides and Web-based content relevant to the concepts of and use of OR techniques.
Outside class, the instructor uses Web-based course management software (for example WebCT, BLACKBOARD) to interact with students as described under course format section.
All skill-building exercise and case studies assigned in this class require students to use software packages available in Microsoft Excel or specialized OR software.
When working on case assignments, each team is set up an on-line discussion group and repository to facilitate collaboration among team members.

D. Special Project

Students are required to apply the interaction design activities to design and implement a mobile application. Each group of students chooses one problem. A prototype and a detailed document showing how the interaction design principles and activities were applied for the design and construction for the mobile application should be submitted.

CLASS SCHEDULE (Tentative)

Week#	Date	Topics covered	Sections to Read
1		What is interaction design	Ch. 1
2		Understanding and conceptualizing interaction	Ch. 2
3		Interfaces and interactions	Ch. 6
4		Data gathering	Ch. 7
5			
6		Break	
7		The process of interaction design	Ch. 9
8		Identifying needs and establishing requirements	Ch. 10
9		Design, prototyping and construction	Ch. 11
10		Introducing evaluation	Ch. 12
11		Usability testing and field studies	Ch. 14
12		User-centered design for mobile apps development	
13		Design patterns for mobile apps	
14		UI design for mobile apps	
15		The Apache Cordova framework	

ITAP 3471: Web Server Administration

Prerequisites

ITAP 2312: Web Development
ITAP 2431: Network Management.

Credit hours

4 Credit Hours

Course overview

The primary objective of this course is to give students a comprehensive overview of the tools and techniques needed to successfully administer Web servers. The course is designed so as to cover topics that are relevant to the role of a Web server administrator. Topics include installation, configuration, and administration of Web servers on common hardware/software platforms.

PMU Competencies and Learning outcomes

This course helps students develop proficiency in the installation, configuration, and administration of commonly used Web server. Students develop both the conceptual basis and the practical skills in the design and implementation of Web server software to support the core and mission-critical Internet-based business processes of an organization. Additionally, this course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course includes a structured laboratory component to ensure that students gain the necessary experience and skill in handling Web server environment. The course includes individual as well as group projects and provides opportunities for the presentation and defense of their designed architectural solution.

Detailed Course Description

The course begins with an introduction to the role of server and Web server in Web environment. It then proceeds with a review of major tasks performed by Web administrators. This sets the stage for discussing in detail the process of installing, configuring, deploying, testing and monitoring a Web server. In learning the installation task, students acquire proficiency in installing various component software (server operating system, Web server, Domain Name Server), setting up IP addressing schemes, and configuring TCP/IP. In configuring Web server, students are exposed to the process for authenticating users, establishing network policies, setting file permissions, and sharing network resources within the Web server environment. The task of securing Web server permits students to understand the role of security and provides hands-on practice in installing and configuring security-related software (firewall). Finally, students have an opportunity to deploy, test and monitor the Web server. This step familiarizes students with installing and supporting various components of an Internet-based information system. Students are exposed to these tasks for the most commonly used Web server software in industry.

Requirements Fulfilled

This course is required for all students majoring in Information Technology in the College of Computer Engineering and Science. It is also recommended as an elective for students majoring in computer science and management information systems. It should be taken no earlier than the first semester of the junior year.

Required Prerequisites

GEIT 1411: Computer Science I
GEIT 1412: Computer Science II
ITAP 2431: Network Management.

Learning Outcomes

In this course, students learn:

- Define the role of Web servers in mission-critical, Internet-based information systems
- Configure the Domain Name server (DNS) and briefly explain the NetBIOS name resolution process.
- Configure Windows Internet Naming Service (WINS) and briefly explain the tasks it performs
- Define the security measures in web server environments and demonstrate the ability to setup a subset of them.
- Install and configure a web server using either Internet Information Server (IIS) or Apache server
- Develop the communication, leadership and teamwork skills necessary to work in teams, or in charge of

teams, that are responsible for operating Web server environments.

Assessment Strategy

The student's performance in this course may be assessed on the basis of:
Two exams, Midterm and one comprehensive final examination.
Laboratory exercises completed during scheduled lab sessions.
One comprehensive final project completed outside of class.
Class participation.

GRADING SCHEME:

Weekly Lab Assessment	20%
H/W & Quizzes	20%
Mid-Term	20%
Class participation	5%
Final project	15%
Final Exam	20%

Course Format

This course utilizes a mix of in-class lectures and student- as well as instructor-led discussions designed to help students develop a deep understanding of the role of ethics and professional conduct in their chosen discipline. While class meetings are utilized to introduce students to the theory of ethical reasoning and process for resolving ethical conflicts, contemporary case studies and "What would you do?" scenarios are used to impart an understanding of guiding principles for ethical and professional conduct. These cases are drawn from the required textbook and supplementary printed material.

Once every two weeks, the instructor assigns one real-world case for students to analyze and respond with a written report. Randomly selected students are asked to make oral presentations of their response.

Once every two weeks and alternating with written case analyses, the instructor posts a "What would you do?" scenario for on-line discussion. An online discussion group is set up to discuss these scenarios outside of the classroom. Students are required to actively participate in this online forum. In addition to contributing to the discussion, the instructor monitors discussion activity for grading purposes. Further, the instructor chooses students at random to present an oral summary of the discussion to the class. This motivates all students to participate in on-line case discussions.

In addition, the instructor should consider creating a Web site for this course using Web technologies such as WebCT or BLACKBOARD. At minimum, the site should include:
Course syllabus
Lecture material (for example, PowerPoint slides, lecture notes, etc.). These should be placed on the site ahead of class meeting so that students may use the material to prepare for the lecture
Summary of cases in ethics that are discussed in class
Out-of-class assignments in the form of mini cases in Ethics
Keys to exams (after students have completed them)
Suggested solution to mini cases (after graded assignments have been returned)
Mechanism for students to digitally submit their assignments
Course calendar
Mechanism to communicate electronically (for example, e-mail)
Discussion groups. Students participate in one on-line case discussion every two weeks.
The student's course performance measures

Topics to be covered

A. Window Server 2012
1. Introduction to Window Server 2012
2. Window Server Edition
3. Desktop changes, Active Directory changes, Networking changes and Management tools

B. Preparing for server installation and configuration
1. Identify server categories and evaluate server components
2. Planning for system disasters and recovery

3. Understand the installation process
4. Install Windows Server 2012 R2
5. Set up IP addressing and configure TCP/IP
6. Install and configure domain name server (DNS)
7. Install and configure IIS

C. Managing a Web server
1. User authentication
2. Manage users, groups, and file system permissions
3. Share resources in a network
4. Enforce network policies

D. Securing a Web server
1. Identify threats and vulnerabilities
2. Secure data transmission, OS, and server applications
3. Authenticate Web users

E. Performance tuning
1. Monitor Web servers and Web applications
2. Analysis tools for Web servers

Laboratory Exercises

Each week, students attend three one-hour scheduled laboratory sessions. During these sessions, students learn the several steps involved in the design, installation, configuration, and administration of a Web server environment. The following administration tasks should be covered in these labs. Because of extensive nature of certain tasks, some of these topics may have to be spread over multiple lab sessions.
Install server operating system
Set up IP addressing and Configure TCP/IP
Install and configure DNS
Install and configure Web server software
Implement user authentication, user groups, and file permissions
Set up and implement network policies
Securing and monitoring Web Server

Technology Component

A. In class, the instructor makes use of state-of-the art multimedia projection equipment and software. These are used to project slides and Web-based content relevant to Web server administration.

B. Outside class, the instructor uses Web-based course management software (for example WebCT, BLACKBOARD) to interact with students as described under course format section.

C. Outside class, in the laboratory setting, the instructor makes use of commercial Web server software (Microsoft and Non-Microsoft) to demonstrate various tasks involved in successfully installing, configuring, and managing a Web server.

Special Projects/Activities

A team project is described in Section VII. Assessment Strategy

Textbooks and Teaching Aids

A. Required Textbook
Mark Minasi, Kevin Greene, Mastering Windows Server 2012 R2,
ISBN: 978-1-118-28942-6

B. Alternative Textbooks
Michael Palmer, MCITP Guide to Microsoft Windows Server 2008, Server Administration, Exam #70-646 (Test Preparation) 1st Edition
ISBN-13: 978-1423902386
ISBN-10: 1423902386

C. Supplemental Print Materials
As available from publisher.

D. Supplemental Online Materials
As available from publisher.

Recent articles and case studies taken from online industry sources such as Microsoft.com and IBM.com. The instructor provides the links to these articles and case studies, which are freely available.

Class attendance

Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Withdrawal

It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

CLASS SCHEDULE (Tentative)

Week#	Lecture (3 hours/week)	Lab (3 hours/week)	Projects/ Readings
Week #1	Syllabus INTRODUCTION What's new in Window Server 2012 R2	Laboratory exercises	Chapter 1
Week #2	Installing and upgrading to Window Server 2012 R2	Laboratory exercises	Chapter2
Week #3	Introduction to Server Core	Laboratory exercises	Chapter 3
Week #4	IP address management and DHCP management	Laboratory exercises	Chapter 5
Week #5	DNS and Name Resolution in Window Server 2012 R2	Laboratory exercises	Chapter 6
Week #6	Active Directory	Laboratory exercises	Chapter 7
Week #7	Creating and Managing User Account	Laboratory exercises	Chapter 8
Week #9	Creating and Managing Shared folder	Laboratory exercises	Chapter 14
Week #10	Web Server Management with IIS		Chapter 19
Week #11	Web Server Management with IIS	Laboratory exercises	Chapter 19
Week #12	creating, using and administering remote desktop services		Chapter 27
Week #13	creating, using and administering remote desktop services	Laboratory exercises	Chapter27
Week #14	Monitoring Window Server 2012	Laboratory exercises	Read Chapter 30
Week #15	Monitoring Window Server 2012	Laboratory exercises	Read Chapter 30

ITAP 3411: Systems Programming

Semester Credit Hours: 4 (3,1)

I. Course Overview

Systems programming is the study of the basic programming principles and skills for building systems software, including the introduction to UNIX, shell programming, C, and Python programming.

II. PMU Competencies and Learning Outcomes

Students in this course develop skills necessary for building systems software over UNIX platform. These skills are necessary for continued success in computer science. This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course includes a structured laboratory component to ensure that students gain the necessary experience and skill in managing the concepts introduced in the class. The course includes individual as well as group projects and provides opportunities for the presentation and defense of designed solutions.

III. Detailed Course Description

COSC 3411: Systems Programming is concerned with the basic programming principles and skills for building systems software, including the introduction to UNIX, shell programming, C, and Python programming. The course presents the students with the concepts of UNIX editor, utilities, file systems, links and shells, shell programming, C, and Python programming. Students are exposed to a variety of techniques for the implementation and uses of systems software. One important lasting effect of this course is to enhance and develop the ability to specify, design, implement and test solutions to programming problems utilizing the data structures and proven algorithms presented in this course.

IV. Requirements Fulfilled

This course satisfies four hours of the requirements for the degree in computer science. It is required of all students pursuing a degree program in computer science within the College of Computer Engineering and Science. It should be taken in the first semester of the junior year.

V. Required Prerequisites

GEIT 3311: Computer Organization

VI. Learning Outcomes

In this course, students learn:

- Describe the Unix environment and applications.
- List and apply Unix commands used in systems administration for file, users, and process management.
- Write Shell and Python scripts for file manipulation and task automation.
- Write C programs in the in the context of systems programming
- Develop improved communication and collaborative skills

VII. Assessment Strategy

This course is designed with three primary goals in mind: to introduce students to the conceptual basis and practical issues associated with the use and development of systems software, to provide students with significant experience in the development of systems software within a profession development environment,

and to provide students with the opportunity to communicate their designs and implementations to their peers in a professional setting. With this in mind, the course grade involves an assessment of their performance on examinations that focus on the application of programming techniques to the solutions of problems and the communication of designed solutions to those problems to an audience. Course grades are based on:

- Weekly assigned homework to motivate students to do the work and earn credit accordingly.
- Weekly structured laboratory exercises designed to guide students through specific course topics.
- Midterm examinations to assess the student's accumulative mastery of content covered prior to the time of the examination.
- Five programming assignments testing students understanding of the major concepts introduced during the course.
- A comprehensive final examination to assess the student's accumulative mastery of course material.

The final grade is based on 25% credit for the programming assignments and project, 5% for the participation in classroom discussions, 20% for weekly laboratory exercises, 20% for midterm examinations, 10% for quizzes, and 20% for the final examination.

Students are required to maintain a journal of thoughts and commentaries during the course. The journal contains daily entries including the identification of areas of interest and concern, notes on the preparation of presentation and comments and analysis of classmate's presentations. The journal is reviewed weekly by the instructor to provide feedback to the students.

Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he proceeds through the curriculum.

Assessment:

Assignments/Projects		10%
Quizzes	15%	
Midterm/Majors	25%	
Final exam	25%	
Lab Work	20%	
Participation		5%

Grading Scale: Standard

A+	96 - 100%
A	90 - 95 %
B+	86 – 89 %
B	80 – 85 %
C+	76 – 79 %
C	70 – 75 %
D+	66 – 69 %
D	60 – 65 %
F	Below 60 %

Class Rules:

Disruptive Behavior
Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.
Eating / Drinking / Smoking: students are requested to refrain from engaging in these activities while in class.

Class attendance
Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness
If a student is late for 3 times it is counted as one absence.

Make-up
Midterm & Final exams – different from and harder than in-class exams. You should a valid excuse as per PMU policies

Late submissions – score reduced by 10% points per day after due date. No work will be accepted after one week of the due date.

Withdrawal
It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Academic Honesty and Integrity
The instructor strictly adheres to all university policies regarding academic integrity.
Academic dishonesty includes but not limited to the following:

- Cheating on examination or other academic work,
- Plagiarism and
- Collusion which means unauthorized collaboration with another in preparing work offered for academic credit.

Academic dishonesty will not be tolerated and the PMU academic regulations will be strictly applied. Unless specifically expressed by the instructor, collaboration between students in this course, between students in previous courses, external assistance in any form or presenting resources/research without proper citation which has been developed by another individual or organization is strictly prohibited. ALL WORK MUST BE THE RESULT OF YOUR OWN EFFORTS.

Miscellaneous
Mobile phones: A student whose mobile rings during class will be asked to leave the classroom and will receive ½ an absence. Should this happen during an exam, the student will not be allowed to retake the exam at another time, while at the same time receiving a full absence.

VIII. Course Format
A. Instruction
This course utilizes both lecture/discussion and laboratory exercises. Students are expected to attend three hours of lecture/discussion per week and two hours of laboratory per week. At least once per week students

should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Once a week students should have at least 30 minutes of collaborative problem solving activity.

B. Web supplement

Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:
Course syllabus
Course assignments
Sample solutions to examinations (after being graded and returned)
Sample solutions to programming assignments (after being graded and returned)
Course calendar (an active utility)
Course e-mail (an active utility)
Course discussion list (an active utility)
Student course performance (an active utility)

Classroom Hours (5 hours per week) Class: 3
Lab: 2

IX. Topics to be covered

A. UNIX/LINUX environment
1. Editor
2. Utilities
3. File systems
4. Links and shells
5. Shell programming

B. C programming
1. Basics
2. Control flow
3. Pointers, arrays, functions
4. File manipulation
5. Process Control

C. Python programming
1. Basics
2. Control flow
3. File manipulation
4. Task Automation
5. Standard Python Modules

X. Laboratory Exercises

This course requires a weekly 2-hour laboratory component. Topics to be covered in the laboratory sessions should include:

Lab 1- Linux Introduction and Ubuntu Installation
Lab 2- Linux commands for Files and directories manipulation
Lab 3 – Linux commands for System Users Management
Lab 4,5 – UNIX Shell Programming practice
Lab 6– Introduction to C Programming
Lab 7,8 – Pointers, memory allocation, and functions in C
Lab 9 – Introduction to Systems programming in C (file manipulation, etc)
Lab 10 - Process Control in C
Lab 11 – Introduction to Python Programming
Lab 12,13 – File manipulation and task Automation using Python

Lab 14 – Modules in Python

Three additional lab sessions should be kept in reserve to allow the instructor to extend the more difficult laboratories for more than one session.

XI. Technology Component

This course makes use of the university's wireless access infrastructure. The course relies on the university and the students having access to professional grade application development environments for the students to use. The course has a laboratory component that would be best implemented in university provided laboratory space.

XII. Special Projects/Activities

Students are required to keep a "reflective notebook" in which, after each class, they enter their own assessments of what they learned, and what questions remain from the class. From each exercise set, each student selects one problem, which the student thinks best reflects the way the topic is used in a technical context. A detailed solution to the problem is included in the student's reflective notebook.

XIII. Textbooks and Teaching Aids

A. Required Textbook

1. Das, S. Your UNIX: The Ultimate Guide. 3rd Edition 2013
   McGraw-Hill, 2001.
   ISBN 0-07-108629-5

2. Learning Python, 5th Edition Fifth Edition Edition
   Publisher: O'Reilly Media; Fifth Edition edition (July 6, 2013)
   ISBN-13: 978-1449355739

B. Alternative Textbooks

None.

C. Supplemental Print Materials

None.

D. Supplemental Online Materials

As available from publishers.

ITAP 3382: Business Intelligence

Prerequisites

GEIT 1411: Computer Science I
GEIT 1412: Computer Science II
GEIT 1311: Computer Organization I
GEIT 3341: Database Design

Credit hours

3 Credit Hours

Course overview

The objective of this course is to give students an understanding of key issues involved in business intelligence applications in organizations. The course is designed so as to cover topics that are relevant from a business intelligence perspective specifically focusing on relevant data mining approaches. It is oriented toward the provision of online access to aggregate data analysis results to a variety of physically distributed organizational users. It includes a mix of lectures (some of which are conducted in the laboratory) and discussions on contemporary articles from industry publications.

PMU Competencies and Learning outcomes

This course helps students develop the ability to become conversant on business intelligence topics and understand the related terms and issues that are important for business intelligence professionals around the world. Additionally, the course provides the students with the relevant data mining approaches used in business intelligence applications. Finally, the course goes beyond looking at business intelligence as a "toy" tool, by imparting on the students an understanding of business intelligence as a key competitiveness-enhancing tool for organizations, both large and small.

Detailed Course Description

The course begins with a discussion of the importance of business intelligence applications and their relationship with data mining. It then proceeds with an introductory discussion of business intelligence applications in organizations. This covers several fundamental topics such as qualitative vs. quantitative data analysis, data warehousing for business intelligence applications, analytical data processing, and data mining. The course also covers key concepts and methods. These include graphical and statistical analyses, time series analyses, cluster analysis, exploratory analysis, simple and multiple regression analyses, and multidimensional scaling. The course concludes with a discussion of advanced issues in connection with business intelligence applications, particularly Web-based applications. The emphasis in this course is more on data collection and analysis issues than on database design or programming issues, whose coverage here is minimal.

Learning Outcomes

In this course, students learn to:

- Explain business intelligence and data warehousing topics and related terminology and issues.
- Develop the required knowledge and skills to be able to apply clustering techniques to identify groups of related data
- Develop the required knowledge and skills to be able to apply classification techniques for making predictions using historical data
- Develop the required knowledge and skills to be able to identify association rules and frequent patterns from the data
- Design an enterprise level data warehouse
- Demonstrate the communication, leadership and teamwork skills necessary for effectively working as professionals in teams, or in charge of teams

Required Text

A. Required Textbook

Stacia Misner, Michael Luckevich, Elizabeth Vitt, Business Intelligence, The Microsoft Press, Reprint Edition, Published 12/10/2008
ISBN: 978-0-7356-4084-9.

B. Alternative Textbook

Norusis, Marija J., PASW Statistics 18, Guide to Data Analysis, Prentice Hall, 2010.
Book and CD-ROM Edition. ISBN 978-0321690586

C. Supplemental Print Materials

SPSS, Inc., SPSS 14.0 for Windows (Student Version), Prentice Hall, 2006.
Book and CD-ROM Edition
ISBN 978-0132283076

D. Supplemental Online Materials

Recent articles taken from online industry publications such as the Intelligent Enterprise and CTO magazines. The instructor provides the links to the articles, which are freely available from the Web.

Class Rules

Disruptive Behavior
Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance
Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non-attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness
When a student is late (5 minutes) for 3 times it is counted as one absent.

Withdrawal
It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W", after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment
Students are assessed based on: (a) their performance in two exams (midterm and final); (b) their quizzes/ assignments; and (c) the quality of a final team project and related oral presentation. The relative weights of each of these items on the final grade are as follows:
Two term exam accounts for 30% and a final exams account for 30% of the grade. Combined, they account for 60% of the grade.
Homework assignments accounts for 20% of the grade. You can expect 3-4 assignments during the course.
Class participation accounts for 5% of the grade, and is evaluated based on the student's active participation in speakers' presentations, and the ability of students to add to the material already provided by the instructor to them
The class project accounts for 15% of the grade. It is evaluated based on a project document, oral presentation, and code quality. The project must be conducted in collaboration with a client organization (for example, a department at a large company or non-profit organization). A letter from the main contact person at the client organization, discussing and evaluating the project and its outcomes, must be provided to the instructor. The letter should contain the contact information of the person writing so the instructor can call him/her up and inquire about the project.

The exams encourage the students to review all of the concepts and methods discussed in class, which are primarily based on textbook material. This is complemented by the class discussions on recent articles taken from online industry publications, which allow the students to become conversant with the industry-specific lingo. The final project provides an experience where concepts, methods, and industry-relevant issues are all brought together in a very applied manner to solve a real problem faced by a real organization. While

this project is not as extensive as a program capstone project, it gives the students the necessary exposure to industry-relevant issues to prepare them for the future challenge of conducting a final program capstone project, and subsequently pursuing a successful career as IT professionals.

Evaluation	
Homework	20 %
Class Participation	5%
Project	15 %
Exam 1	15 %
Exam 2	15 %
Final Exam	30 %
Total 100%	

Grading Scale

A+	96 - 100%
A	90 - 95 %
B+	86 – 89 %
B	80 – 85 %
C+	76 – 79 %
C	70 – 75 %
D+	66 – 69 %
D	60 – 65 %
F	Below 60 %

Course Format

A. Instruction

This course is based on lecture/discussion inside the class. Students are expected to attend three hours of lecture/discussion per week.

CLASS SCHEDULE

Week	Chapters	Topics
1	Class Introduction/ Chapter 1	Course Contents, Introduction to Data Mining
2-3	Chapter 2	Knowing your data
4	Chapter 10	Cluster Analysis
5	Chapter 10	Cluster Analysis
6	Chapter 6	Introduction to Rapid Miner/Mining Frequent Patterns
7	Chapter 6	Mining Frequent Patterns
8-9	Chapter 4	Data Warehousing and OLAP
10	Chapter 4	Data Warehousing and OLAP
11-12	Chapter 3	Data Preprocessing
13-14	Chapter 8	Classification
15		Advanced Topics, Project Presentation, Course Review

ITAP 3383: Enterprise Resource Planning Systems

Semester Credit Hours: 3 (3,0)

I. Course Overview

The course is intended to introduce students to the underlying need for information and business process integration in large organizations. It takes an overview look at the functional areas of a business and the business processes that support the functional areas. The emphasis is on the use of advanced information technology for integrating business functions through distributed databases for support of internal business functions. It includes a discussion of the idea behind selection and implementation of enterprise resource planning (ERP) systems. A part of the course is set aside for demonstrations and "hands on" exercises with one of the available ERP software. Students use this software to perform some of the processes and tasks to create, track, and communicate enterprise information.

II. PMU Competencies and Learning Outcomes

This course introduces students to the design and development of enterprise resource planning systems. Students develop both the conceptual basis and the practical skills in the design and implementation of enterprise-wide information systems to support the core and mission-critical business processes of an organization. They gain an understanding of and appreciation for the need for integrating business processes and information in large organizations for the purpose of not only planning and control but also to gain strategic competitive advantage. Additionally, this course makes an extensive use of the PMU technology infrastructure to provide communication between faculty and students. While the course does not include a structured laboratory component, out-of-class assignments are assigned to ensure students gain an introductory experience in working with an ERP system. The course includes group projects and provide opportunities for the presentation and defense of their designed solution.

III. Detailed Course Description

The course is designed to provide an overview of enterprise resource planning systems. In particular, the course focuses on the ERP life cycle. The course introduces students to problems that traditionally fragmented information systems create and therefore, to the underlying need for integration of business processes and information in large organizations. Further, the course focuses on implementation difficulties inherent in installing such systems and permit students to become knowledgeable about the issues and challenges in introducing such systems in large organizations. A part of the course is set aside for demonstrations and "hands on" exercises with one of the available ERP software. Students use this software to perform some of the processes and tasks to create, track, and communicate enterprise information.

The course covers three areas, which are somewhat interleaved during the term.
The first area acquaints students with the conceptual problems of integration in organizations, the organizational characteristics that tend to make these problems more or less severe, and possible courses of action for firms faced with severe enough problems that they wish to take action. Attention is focused on ERP systems as a particularly interesting possible solution.
The Second part of the course gives students an overview of one particular ERP system (SAP is strongly recommended). This portion of the course covers overall architecture, the user interface, and the major points of integration between its modules. Students learn how to enter transactions on the system, and to track the interactions and information flows between different business functions.
The third part of the course focuses on how to implement an integrated enterprise-wide system into a large organization.

A running case study is used to illustrate above concepts. The mastery of these concepts and skills is demonstrated via an end-of-term team project. This course may be taught using one of the many ERP software available such as SAP, PeopleSoft, Oracle, or J.D. Edwards. SAP is highly recommended for several reasons. For one, it is the leading ERP software in use today. Secondly, SAP offers a program through which academic institutions can acquire SAP software and sample database at little or no cost. Universities that do not wish to install their own instance of SAP can use a hosted instance at one of the many universities worldwide. Finally, textbooks are easily available for SAP then other ERP software.

IV. Requirements Fulfilled

This course is required for all students majoring in Information Technology in the College of Computer Engineering and Science. It is also recommended as an elective for students majoring in computer science and management information systems. It should be taken no earlier than the first semester of the junior year.

V. Required Prerequisites

GEIT 1411: Computer Science I
GEIT 1412: Computer Science II
GEIT 2341: Database Design.

VI. Learning Outcomes

In this course, students learn:

- Become familiar with the need for integrating business processes.
- List, Define and Differentiate the concepts of enterprise-wide resource planning systems.
- Name and contrast the many available software packages used to implement an ERP solution.
- Describe the elements and process related to the design and development of various enterprise information system modules using the selected ERP software.
- List risk factors and success ingredients relevant to the design and implementation of enterprise-wide resource planning systems.
- Show communication, leadership and teamwork skills necessary to work in teams, or in charge of teams, that are responsible for implementing mission-critical ERP solutions.

VII. Assessment Strategy

Students are assessed based on: their performance on two exams (midterm and final); their class participation, which includes the discussion of recent articles taken from online industry publications; and the quality of a final team project and related oral presentation. The relative weights of each of these items on the final grade are as follows:

Evaluation

Class participation		5 %
Quiz		10 %
Homework		5 %
Project		20 %
Exam1	15 %	
Exam2		20 %
Final Exam		25 %
Total 100%		

VIII. Course Format

This course utilizes a mix of in-class lectures, discussions, and software demonstrations designed to help students learn the various tasks involved in designing and implementing an ERP solution. While class meetings are utilized to emphasize conceptual foundation in topics related to these tasks, software demonstrations are used to provide students with hands-on training in performing these tasks. For best results, a part of this course should be taught in a lab-environment where students can follow the instructor during software demonstrations and discussion of key development strategies. Initial class meetings introduce students to the theoretical concepts. Most of the later meetings are used for laboratory software demonstrations geared at helping students master skills for designing and developing various parts of an ERP system. A running case study is used for this purpose so that students see an end-to-end development cycle. Students are expected to attend three hours of lecture per week. There are no scheduled lab hours for this course.

In addition, the instructor should consider creating a Web site for this course using Web technologies such as WebCT or BLACKBOARD. At minimum, the site should include:
Course syllabus
Lecture material (for example PowerPoint slides, lecture notes, etc.). These should be placed on the site ahead of class meeting so that students may use the material to prepare for the lecture
Software demonstration exercises completed in class
Out-of-Class assignments and end-of-term project
Keys to exams (after students have completed them)
Solution to Laboratory Exercises (after graded assignments have been returned)
Mechanism for students to digitally submit their assignments
Course calendar
Mechanism to communicate electronically (for example e-mail)
Discussion groups
Students course performance measures

Classroom Hours (3 hours per week) Class: 3

Lab: 0

IX. Topics to be Covered

A. ERP concepts

1. Evolution of information systems
2. Emergence of ERP software

B. Business functions, process and data requirements

1. Functional areas and business processes
2. Functional area information systems

C. Planning, design, and implementation of ERP systems

1. ERP data inputs
2. ERP output capabilities
3. Selecting an ERP solution
4. Designing an ERP solution
5. Implementing an ERP solution
6. Manage the "go live" phase

D. ERP system in a large organization

1. ERP modules for sales and marketing
2. ERP modules for accounting and finance
3. ERP modules for production and materials management
4. Complete ERP solution

E. ERP and e-commerce: using ERP through the Web

X. Laboratory Exercises

Students are assigned one out-of-class application development exercise every three weeks. These hands-on exercises are expected to be done in a team setting (generally 3-4 students/team) and are designed to illustrate various development concepts covered during lecture meetings. Thus, students are expected to complete about five such exercises. Collectively, these exercises enable students to learn the several steps involved in the design and implementation of a relatively simple ERP application. The following major areas should be covered in these exercises:

- Identify and design various modules for an ERP solution
- Design, and develop sales order processing and distribution module
- Design and develop production planning process module
- Design and develop an accounting module to generate financial statements and product profitability analyses reports
- Design and develop a human resource module to manage employee benefits

XI. Technology Component

A. In class, the instructor makes use of state-of-the art multimedia projection equipment and software. These are used to project slides and Web-based content relevant to the concepts of enterprise resource planning systems.

B. In class, in a laboratory setting, the instructor makes use of a commercial ERP software to demonstrate various tasks involved in successful design and implementation of an enterprise-wide resource planning system.

C. Outside class, the instructor uses Web-based course management software (for example WebCT, BLACKBOARD) to interact with students as described under course format section.

XII. Special Projects/Activities

An end-of-term team project is required in this course. This project is designed to permit students to apply concepts, methods, and tools learned in class to develop an ERP solution for a fictitious company. The project is require students, working in teams, to design and implement a complete ERP solution using ERP software such as SAP. Student teams are expected to:
Design, test, and deploy the application,
Prepare a complete technical manual, and
Make an oral presentation of their solution.

The final project is thus designed to assess competency in performing various tasks related to the conceptual design and implementation of an ERP system.

XIII. Textbooks and Teaching Aids

A. Required Textbook

Ellen Monk, Bret Wagner; Concepts in Enterprise Resource Planning, 4th Edition, 2013, Cengage, ISBN-13: 9781111820398

B. Alternative Textbook

O'Leary, Daniel E., Enterprise Resource Planning Systems: Systems, Life Cycle, Electronic Commerce, and Risk, Cambridge University Press, 1st Edition, 2000.
ISBN 0521791529

C. Supplemental Print Materials

As available from publisher.

D. Supplemental Online Materials

As available from publisher.

ITAP 4371: E-Commerce

Prerequisites

GEIT 1411: Computer Science I
GEIT 1412: Computer Science II
GEIT 2341: Database I
ITAP 2312: Web Programming

Credit hours

3 Credit Hours

Course overview

The primary objective of this course is to expose students to the advanced use of information technology in the design and implementation of Web-based business applications to support e-commerce. The course presents concepts, methodology, and tools for designing, implementing, and management of e-commerce applications.

PMU Competencies and Learning outcomes

This course helps students develop proficiency in the design and development of e-commerce applications. Students develop both the conceptual basis and the practical skills in the design and implementation of Web-based applications to support the core and mission-critical Internet-based business processes of an organization. Additionally, this course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. While the course does not include a structured laboratory component, out-of-class projects are assigned to ensure students gain the necessary experience and skill developing e-commerce applications. The course includes a group project and provides opportunities for the presentation and defense of their designed solution.

Detailed course description

The course is designed to provide coverage of technologies available to design, develop, implement, and manage business information processing applications that support e-commerce. The course begins with an overview of E-Commerce concepts and technologies. It then presents client-side web programming languages such as HTML5 and CSS3. It then introduces students to the use of an integrated development environment based on J2EE and/or .Net framework (for example IBM WebSphere Studio or Visual Studio .Net) in the development of various components that make up an E-Commerce application. Next, students learn to develop server-side components using technologies such as ASP.net, JavaBeans, Java Server Pages. Connectivity of the server-side components to back-end databases is introduced next. Finally, the course presents the model-view-controller (MVC) design paradigm.. The mastery of these concepts and skills is demonstrated via an end-of-term team project.
This course may be taught using one of the two competing object-oriented technologies, namely industry-standard J2EE platform or Microsoft's .Net framework. The choice of technology should depend largely on student's programming background (Java versus Visual basic or C#), industry demands, and availability of faculty with requisite skills. For these reasons, the topical outline is left generic enough to be compatible with either technology. J2EE is currently the most commonly used framework for e-commerce applications. However, Microsoft's .Net framework is gaining momentum.

Requirements fulfilled

This course is required for all students majoring in Information Technology in the College of Computer Engineering and Science. It is also recommended as an elective for students majoring in computer science and management information systems. It should be taken no earlier than the first semester of the senior year.

Learning Outcomes

In this course, students learn:

CLO1 To demonstrate knowledge about the design, development and implementation of client-side components that constitute an e-commerce web application
CLO2 To employ frameworks (for example J2EE or .Net) to design, develop and implement server-side components that constitute an e-commerce web application
CLO3 To develop skills in using an integrated development environment (IDE) for example IBM WebSphere

or MS Visual Studio .Net

CLO4 To analyze main issues relevant to the implementation of enterprise-level e-commerce web applications

CLO5 To show communication, leadership and teamwork skills necessary to work in or lead teams

Class Rules

Disruptive Behavior

Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.

Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance

Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness

When a student is late for 3 times it is counted as one absent. Students are considered tardy if they arrive after the first 10 minutes (according to the instructor's watch).

Withdrawal

It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W", after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment

There will be individual homework assignments as well as one group project broken up into several phases. Some of the assignments will require programming. There will also be frequent in-class discussions and activities in which students are expected to actively participate. There will be a midterm and a final exam.

The final grade will be calculated this way:
Programing assignments and presentations: 15%
Quizzes: 10%
Major 1: 15%
Major 2: 15%
Group project: 20%
Final: 25%

Course Format

This course utilizes a mix of in-class lectures, discussions, and software demonstrations designed to help students learn the various tasks involved in designing, installing, configuring, securing, and monitoring an e-commerce application. While class meetings are utilized to emphasize conceptual foundation in topics related to these tasks, software demonstrations are used to provide students with hands-on training in performing these tasks. For best results, this course is taught in a lab-environment where students can follow the instructor during software demonstrations and discussion of key development strategies. Initial class meetings introduce students to the theoretical concepts. Most of the later meetings are used for laboratory software demonstrations geared at helping students master skills for designing and developing various E-Commerce application components. A running case study is used for this purpose so that students see an end-to-end development cycle. Students are expected to attend three hours of lecture (includes discussion/software demonstrations) per week.

In addition, the instructor should consider creating a Web site for this course using Web technologies such as WebCT or BLACKBOARD. At minimum, the site should include:

Course syllabus

Lecture material (for example PowerPoint slides, lecture notes, etc.). These should be placed on the site ahead of class meeting so

that students may use the material to prepare for the lecture
Out-of-Class assignments and end-of-term project
Keys to exams (after students have completed them)
Mechanism for students to digitally submit their assignments
Course calendar
Mechanism to communicate electronically (for example e-mail)
Discussion groups

Topics to be covered

A. Developing client-side components
   1. Introduction to HTML5
   2. HTML5 elements
   3. HTML5 tables and forms elements

B. Formatting web pages using CSS3

C. Development of server-side components
1. Introduction to ASP.Net
2. Visual Studio Express for Web
3. ASP.Net server Controls
4. ASP.Net validation controls
5. ASP.Net and Database connectivity (ADO.Net)

D. Test, debug and deploy an E-commerce application
1. Test, debug and deploy D.
2. Unit testing
3. Debugging tools
4. Deployment environment to include Web and application servers

E. Securing e-commerce application
1. Identify threats and vulnerabilities
2. Understand various aspects of security such as authentication and authorization.
3. Test and deploy secured application

Technology component

A. In class, the instructor makes use of state-of-the art multimedia projection equipment and software. These are used to project slides and Web-based content relevant to the design of E-Commerce applications.
B. In class, in the laboratory setting, the instructor makes use of a commercial integrated development environment based on J2EE or .Net framework to demonstrate various tasks involved in successful design and development of an enterprise-level, mission-critical E-Commerce application.
C. Outside class, the instructor uses Web-based course management software (for example WebCT, BLACKBOARD) to interact with students as described under course format section.

Special projects/Activities

An end-of-term team project is required in this course. This project is designed to permit students to apply concepts, methods, and tools learned in class to support a real system. The project requires students, working in teams, to design and implement a complete Web-based, n-tiered e-commerce application of reasonable complexity. This may be a simulated application designed to expose students to tasks involved in assembling a real-system or it may be a real application for a real client. In either case, student teams are expected to:
Design, test, and deploy the application,
Prepare a complete technical manual, and
Make an oral presentation of their design architecture to a panel of judges comprising of the instructor and other faculty members or client.

The final project is thus designed to assess competency in performing various tasks related to the design, development and implementation of a complete E-Commerce application.

Required Text

A. Required Textbook

1. If the course is based on .Net framework:
   Imar Spaanjaars; Beginning ASP.NET 4.5.1: in C# and VB; Wrox; 2014
   ISBN: 978-1-118-84677-3 (Available on PMU's bookstore)
2. If course is based on J2EE platform:
   TBA

B. Alternative Textbooks

1. If the course is based on J2EE platform:
   TBA
2. If the course is based on .Net framework:
   Jason N. Gaylord, Christian Wenz, Pranav Rastogi, Todd Miranda,
   Scott Hanselman; Professional ASP.NET 4.5 in C# and VB;
   John Wiley & Sons, 2013
   ISBN: : 978-1-118-31182-0

ITAP 4316: Introduction to Software Project Management

Semester Credit Hours: 3 (3,0)

I. Course Overview

This course will introduce the concepts and techniques for managing software projects. The students will learn the basic project management concepts including process groups, knowledge areas and project management lifecycles with special focus on Agile approaches. The students will learn techniques for initiating, planning, launching and monitoring software projects. They will also learn tools and techniques related to software configuration and build management. They will learn software project management tools and use it to manage their software project.

II. PMU Competencies and Learning Outcomes

This course helps students develop the ability to become conversant with software project management topics and understand the related terms and issues that are important for software engineering practitioners around the world. Additionally, the course provides the students with the communication, leadership, and teamwork skills necessary to effectively work as professionals in teams, or in charge of teams, responsible for software engineering projects. It introduces the students to software project management tools and its applications. It provides students with an understanding of software configuration management and software build management issues.

III. Detailed Course Description

The course introduces the project management topics that are relevant in the context of software engineering projects. This covers several fundamental project management issues such as task decomposition and related budgeting, project charting, project scope management, time and cost management, human resources management, communication management, and project risk management. The course also includes a discussion of advanced issues in connection with software engineering projects, such as emerging tools for software engineering (state of the art project, configuration and build management tools are introduced). This course emphasizes both software design/implementation issues and software engineering project management issues. This dual emphasis orientation is aimed at providing students with a realistic view of issues related to real software engineering projects, which are more often than not complex collaborative projects with a clear expectation of organizational impacts in terms of quality, productivity and/or competitiveness enhancements.

IV. Requirements Fulfilled

This course is elective for all students majoring in Information Technology, Computer Science and Computer Engineering departments in the College of Computer Engineering and Science.

V. Required Prerequisites

GEIT 1411: Computer Science I
GEIT 1412: Computer Science II
GEIT 3351: Software Engineering I

VI. Learning Outcomes

In this course, students learn to:

- Explain and justify the concepts and terms related to software project management
- Select, apply and analyse the appropriate software development life cycle for managing a project
- Design a project plan and schedule
- Develop an effective reporting system for monitoring and controlling the project progress
- Explain the mechanism used for software configuration management and bug/feature tracking
- Develop the communication, leadership and teamwork skills necessary to effectively work as professionals in teams, or in charge of teams

VII. Assessment Strategy

Students are assessed based on: (a) their performance in two exams (midterm and final); (b) their quizzes; and (c) the quality of a final team project and related oral presentation. The relative weights of each of these items on the final grade are as follows:
There will be one midterm exam accounting for 20% and final exams account for 30% of the grade. Combined, they account for 50% of the grade.
Class quizzes account for 20% of the grade.
Class participation accounts for 5% of the grade, and is evaluated based on the student's active participation in speakers' presentations, and the ability of students to add to the material already provided by the instructor to them
The class project accounts for 25% of the grade. It is evaluated based on a project document, oral presentation, and client perceptions of the team project. The project must be conducted in collaboration with a client organization (for example, a department at a large company or non-profit organization). A letter from the main contact person at the client organization, discussing and evaluating the project and its outcomes, must be provided to the instructor. The letter should contain the contact information of the person writing so the instructor can call him/her up and inquire about the project.

The exams encourage the students to review all of the concepts and methods discussed in class, which are primarily based on textbook material. This is complemented by the class discussions on recent articles taken from online industry publications, which allow the students to become conversant with the industry-specific lingo related to software engineering issues. The final project provides an experience where concepts, methods, and industry-relevant issues are all brought together in a very applied manner to solve a real problem faced by a real organization. While this project is not as extensive as a program capstone project, it gives the students the necessary exposure to industry-relevant issues to prepare them for the future challenge of conducting a final program capstone project, and subsequently pursuing a successful career as IT professionals.

Evaluation

Class Participation	5%
Project	25 %
Quizzes	20%
Midterm Exam	20 %
Final Exam	30 %
Total	100%

Grading Scale

A+	96 - 100%
A	90 - 95 %
B+	86 – 89 %
B	80 – 85 %
C+	76 – 79 %
C	70 – 75 %
D+	66 – 69 %
D	60 – 65 %
F	Below 60 %

VIII. Course Format

This course utilizes a mix of in-class lectures, discussions, and software demonstrations designed to help students learn the various tasks involved in managing, designing and implementing a large scale software project. While class meetings are utilized to emphasize conceptual foundation in topics related to these tasks, software demonstrations are used to provide students with hands-on training in performing these tasks. Students are expected to attend three hours of lecture per week. There are no scheduled lab hours for this course.

In addition, the instructor should consider creating a Web site for this course using Web technologies such as WebCT or BLACKBOARD. At minimum, the site should include:
Course syllabus
Lecture material (for example PowerPoint slides, lecture notes, etc.). These should be placed on the site ahead of class meeting so that students may use the material to prepare for the lecture
Software demonstration exercises completed in class
Out-of-Class assignments and end-of-term project
Mechanism for students to digitally submit their assignments
Course calendar
Mechanism to communicate electronically (for example e-mail)
Discussion groups
Students course performance measures

Classroom Hours (3 hours per week) Class: 3

Lab: 0

IX. Topics to be covered

Week	Chapters	Topics
1	Introduction/ Chapter1	What Is a Project? What is Project Management?
2-3	Chapter 2	Understanding The Project Management Process Groups
4	Chapter 3	How to Scope a Project
5-6	Chapter 4	How to Plan a Project/ Introduction to Redmine
7-8	Software Configuration Management Tool	Introduction to Subversion (Midterm Exam)
9	Bug/Feature Tracking	Introduction to Bug/Feature Tracking
10	Chapter 5	How to Launch a Project
11	Chapter 6	How to Monitor and Control a Project
12	Chapter 8	Project Management Landscape
13	Chapter 7	How to Close a Project
14-15	Project/Review	Project Presentations, Course Review/Advanced Topics
16	Final Exam	(Final exam. Scheduled by the Registrar)

X. Laboratory Exercises

There is no lab component for this course. However, students are assigned one out-of-class application development exercise every three weeks. These hands-on exercises are expected to be done in a team setting (generally 3-4 students/team) and are designed to illustrate various development concepts covered during lecture meetings. Thus, students are expected to complete about four such exercises. Collectively, these exercises enable students to learn the several steps involved in the scoping, planning, launching and monitoring the project using an online project management tool (e.g. Redmine). The following major areas should be covered in these exercises:

- Identify project requirements
- Develop RBS and WBS of the requirements
- Develop software design documents (class diagram, ER diagram)
- Develop a plan for the project
- Monitor project progress and obtain weekly status reports

XI. Technology Component

A. In class, the instructor makes use of state-of-the art multimedia projection equipment and software. These are used to project slides and Web-based content relevant to the concepts of software project management

B. In class, in a laboratory setting, the instructor makes use of a open source project management software to demonstrate various tasks involved in successful design and implementation of software systems.

C. Outside class, the instructor uses Web-based course management software (for example WebCT, BLACKBOARD) to interact with students as described under course format section.

XII. Special Projects/Activities

An end-of-term team project is required in this course. This project is designed to permit students to apply concepts, methods, and tools learned in class to manage software system for a fictitious company. The projects require students, working in teams, to scope, plan, design and implement a complete software system. Student teams are expected to:
Gather and document requirements
Develop project plan and project schedule
Design, test, and deploy the application,
Make an oral presentation of their solution.

The final project is thus designed to assess competency in performing various tasks related to the software project management and its implementation.

XIII. Textbooks and Teaching Aids

A. Required Textbook

Effective Project Management: Traditional, Agile, Extreme, 5th Edition
Robert K. Wysocki ISBN: 978-0-470-42367-7 April 2009

B. Alternative Textbooks

C. Supplemental Print Materials
INFORMATION TECHNOLOGY PROJECT MANAGEMENT, 6th Edition Kathy Schwalbe
Managing and Leading Software Projects Richard E. (Dick) Fairley ISBN: 978-0-470-29455-0, March 2009

D. Supplemental Online Materials
As available from publisher.

Instructor Details
Name: Dr. Majid Ali Khan
Semester: Fall 2016-2017
Room: S-016
Tel: +966+3+849-8503
E-mail:makhan@pmu.edu.sa
Office Hours: TBA

Class Rules

Disruptive Behavior
Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance
Class attendance is required and a class roll will be taken during each class period. Learning is an active

process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non-attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness
When a student is late (5 minutes) for 3 times it is counted as one absent.

Withdrawal
It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W", after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

ITAP 3371: Database II

Prerequisites
GEIT 3341: Database I

Credit hours
3 Credit Hours

Course overview

This course is a continuation of GEIT 3341 (Database I) and covers more advanced topics in database systems including advanced SQL, query processing & optimization, transaction processing, concurrency control, database recovery, database security & authorization, object-relational databases, and distributed databases.

PMU Competencies and Learning outcomes

This is an advanced course in database systems. It aims at broadening students' database knowledge by covering more advanced topics. In addition to the basic and intermediate SQL acquired in Database I, the course extends students' SQL skills by covering more advanced features in SQL. Moreover, the course provides the students with an understanding of how queries are processed and optimized within the database management system. To help the students understand how transactions are processed in a multi-user environment, relevant topics including transaction processing, concurrency control, and recovery techniques are covered. Furthermore, students learn how to protect the security of a database against unauthorized access, create and manipulate object-relational and distributed databases. Finally, the course provides the students with the communication, leadership and teamwork skills necessary to effectively work as professionals in teams, or in charge of teams, responsible for developing database application programs.

Detailed Course Description

The course begins by covering advanced SQL features including interaction with Java programs, simple and searched case, rank, running totals, percent to total, and sequences. Following this is a discussion of query processing and optimization where the steps involved in query processing, and how heuristics are used in query optimization, are covered. Topics related to transaction processing are then covered including transaction and system concepts, desirable properties of transactions, schedule types, and characterizing schedules based on serializability. Lock-based concurrency control techniques are discussed next including binary, shared/exclusive, and two-phase. Following this is a discussion of recovery techniques in single and multi-user settings. Next, are topics related to database security and authorization including types of security, discretionary access control based on granting and revoking of privileges, and types of security issues. Object-relational concepts are covered next including user-defined types and functions, nested tables and Varrays. Finally, the course concludes with a discussion of distributed databases including data fragmentation, replication and allocation, and types of distributed database systems.

Learning Outcomes

In this course, students learn:

- Describe, Contrast and Apply advanced DDL and DML SQL
- Describe and Contrast query processing steps and optimization techniques
- Describe transaction processing and concurrency control techniques
- Describe and Contrast database security and recovery techniques
- Describe, Contrast and Apply object-relational databases
- Describe and Contrast distributed database concepts and techniques
- Download, Install and Configure a professional database system, e.g., Oracle or MySQL
- Develop the communication, leadership and teamwork skills necessary to work in, or in charge of teams, to develop database applications in a high level programming language like Java

Required Textbook

Ramez Elmasri and Shamkant Navathe , Fundamentals of Database Systems, Pearson; 7 edition (June 18, 2015)
Language: English
ISBN-10: 0133970779
ISBN-13: 978-0133970777

For the lab component of the course

Steven Feuerstein; Oracle PL/SQL Programming; 6th edition (February 16, 2014); O'Reilly Media
ISBN-10: 1449324452

Class Rules

Disruptive Behavior
Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance
Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non-attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness
When a student is late for 3 times it is counted as one absent.

Withdrawal
It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment
Students are assessed based on: their performance in two exams (midterm and final); their class participation, which includes programming assignment, discussion of recent articles taken from online industry publications; and the quality of a final team project and related oral presentation. The relative weights of each of these items on the final grade are as follows:

GRADING SCHEME:

Participation	5%
Quizzes	5%
Lab	
Individual Labs:	10%
Midterm Lab Exam:	5%
Final Lab Exam:	5%
Midterm	20%
Project	20%
Final	30%
Total	100%

Course Format

A. Instruction

Four of the course's class meetings are used for laboratory demonstrations and activities geared at helping the students learn the several steps involved in designing and implementing a database system. The other class meetings are split into two main components: lectures, and class discussions. The lectures cover several topics outlined later in this syllabus. The class discussions are based on recent articles taken from online industry publications such as the Searchers and CIO magazines, which are freely available from the Web. The instructor provides the links to the articles, which are then downloaded by the students and read prior to class. In class, the students discuss the articles in small teams for about 20 minutes, developing three provocative questions per team. This is followed by a discussion involving the whole class, where each team asks one of the questions they developed, and other teams answer them, until all teams asked at least one of their questions. This discussion format is likely to lead to lively debate on topics that are directly addressed by the article, as well as on topics that are indirectly related to the article.

B. Web supplement

Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:
Course syllabus
Course assignments
Sample solutions to examinations (after being graded and returned)
Sample solutions to programming assignments (after being graded and returned)
Course calendar (an active utility)
Course e-mail (an active utility)
Course discussion list (an active utility)
Student course performance (an active utility)

CLASS SCHEDULE (Tentative)

Week	Chapters	Topics
1,2	Chapter 9	Advanced SQL
3,4	Chapter 22	Object-Relational and Extended-Relational Systems
5,6	Chapter 15	Query Processing Query Optimization
7,8,9	Chapter 17	Transaction Processing: Concepts & Theory
10,11	Chapter 18	Concurrency Control Techniques
12	Chapter 19	Database Recovery Techniques
13	Chapter 23	Database Security
14	Chapter 25	Distributed Databases and Client-Server Architectures
15	REVISION	

Project Samples

Sample #	Title
1	Music Sales Database Application
2	Inventory Control Management Database Application
3	Hospital Management Database Application
4	Library Management Database Application
5	Payroll Management Database Application

Lab Plan

Lab #	Title
0	PreLab
1	Advanced SQL – Simple & Searched CASE
2	Advanced SQL – Ranking
3	Advanced SQL – Sequences
4	Advanced SQL – Running Totals
5	Advanced SQL – Percent to Total
6	Java Oracle Interaction
7	Object-Relational: User Defined Types & Functions
8	Object-Relational: Nested Tables
9	Object-Relational: Varrays
10	IMP & EXP Utility in Oracle

ITAP 4313: Advanced Visual Programming

Prerequisites

ITAP 3313: Human-Computer Interaction
GEIT 3341: Database

Credit hours

3 Credit Hours

Office and Phone

Room: S-020
Phone: 03-849-9224
E-mail: ijenhani@pmu.edu.sa
Office Hours: by appointment

Course overview

The primary objective of this course is to provide students with in-depth knowledge in the development of applications using C# .NET technology. It is designed to show how to analyze problems, design solutions, and implement an advanced Windows application using C# and Visual Studio.

PMU Competencies and Learning outcomes

This course helps students develop proficiency in the design and development of advanced Visual desktop applications. Students develop both the conceptual basis and the practical skills in the design and implementation of desktop applications to support the core and mission-critical business processes of an organization. Additionally, this course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. While the course does not include a structured laboratory component, out-of-class projects are assigned to ensure students gain the necessary experience and skill developing desktop applications. The course includes a group project and provides opportunities for the presentation and defense of their designed solution.

Detailed course description

The course is designed to provide coverage of technologies available to design, develop, implement, and manage advanced visual desktop applications that support businesses of organizations. The course starts with a review of C# syntax and constructs. It then introduces students to the use of an integrated development environment based on .Net framework (Visual Studio .Net) in the development of various components that make up an advanced visual desktop application. The mastery of these concepts and skills is demonstrated via an end-of-term team project.

Requirements fulfilled

This course is recommended as an elective course for all students majoring in Information Technology in the College of computer Engineering and Science It should be taken no earlier than the first semester of the senior year.

Learning Outcomes

In this course, students learn:
Recognize the concepts of layered architecture and justify their uses for professional business problems.
To demonstrate the ability to apply concepts of layered architecture in developing professional solutions, e.g. Windows applications using the .NET Framework including data-driven solutions
Become familiar with more advanced programming concepts including reflection, generics, dynamic SQL and LINQ
Develop skills and apply advanced IDE features, e.g., Build and Compile, Debug, Call Stacks, Break Points and Conditional Execution.
Demonstrate communication, leadership and teamwork skills necessary to work in or lead teams

Class Rules

Disruptive Behavior
Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing

is never allowable, just as disruptive behavior is grounds for dismissal.
Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance
Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness
When a student is late for 3 times it is counted as one absent. Students are considered tardy if they arrive after the first 10 minutes (according to the instructor's watch).

Withdrawal
It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W", after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment
There will be individual programming assignments as well as one group project. There will also be frequent in-class discussions and activities in which students are expected to actively participate. There will be a midterm and a final exam.
The final grade will be calculated this way:
Programing assignments: 15%
Quizzes: 10%
Midterm: 25%
Group project: 20%
Final: 30%

Course Format

This course utilizes a mix of in-class lectures, discussions, and software demonstrations designed to help students learn the various tasks involved in designing, installing, configuring, securing, and monitoring an e-commerce application. While class meetings are utilized to emphasize conceptual foundation in topics related to these tasks, software demonstrations are used to provide students with hands-on training in performing these tasks. For best results, this course is taught in a lab-environment where students can follow the instructor during software demonstrations and discussion of key development strategies. Initial class meetings introduce students to the theoretical concepts. Most of the later meetings are used for laboratory software demonstrations geared at helping students master skills for designing and developing various E-Commerce application components. A running case study is used for this purpose so that students see an end-to-end development cycle. Students are expected to attend three hours of lecture (includes discussion/software demonstrations) per week.

In addition, the instructor should consider creating a Web site for this course using Web technologies such as WebCT or BLACKBOARD. At minimum, the site should include:
Course syllabus
Lecture material (for example PowerPoint slides, lecture notes, etc.). These should be placed on the site ahead of class meeting so that students may use the material to prepare for the lecture
Out-of-Class assignments and end-of-term project
Keys to exams (after students have completed them)
Mechanism for students to digitally submit their assignments
Course calendar
Mechanism to communicate electronically (for example e-mail)
Discussion groups

Topics to be covered

Introduction to C# and .NET framework (Syntax review)
The Visual Studio Community 2015 IDE
Introduction to LINQ and the List Collection
GUI: Windows Forms (Part1)

GUI: Windows Forms (Part2)
Generics
Collections: Generic and Nongeneric collections
Databases and LINQ
Windows 8 UI and XAML OR GUI with Windows Presentation Foundation
Asynchronous programming (async and await)

Technology component

A. In class, the instructor makes use of state-of-the art multimedia projection equipment and software. These are used to project slides and Web-based content relevant to the design of windows applications.
B. In class, in the laboratory setting, the instructor makes use of a commercial integrated development environment based on .Net framework to demonstrate various tasks involved in successful design and development of windows desktop applications.
C. Outside class, the instructor uses Web-based course management software (for example WebCT, BLACKBOARD) to interact with students as described under course format section.

Special projects/Activities

An end-of-term team project is required in this course. This project is designed to permit students to apply concepts, methods, and tools learned in class to support a real system. The project requires students, working in teams, to design and implement a complete Windows desktop application of reasonable complexity. This may be a simulated application designed to expose students to tasks involved in assembling a real-system or it may be a real application for a real client. In either case, student teams are expected to:

- Design, test, and deploy the application,
- Prepare a complete technical manual, and
- Make an oral presentation of their design architecture to a panel of judges comprising of the instructor and other faculty members or client.

The final project is thus designed to assess competency in performing various tasks related to the design, development and implementation of a complete visual .Net application.

Required Text

A. Required Textbook

Paul Deitel, Harvey Deitel; Visual C# 2012 How to Program (5th Edition); Deitel Series; 2014.
ISBN: 978-0273793304

B. Supplement Textbook

Adam Nathan; WPF 4.5 Unleashed 1st Edition; SAMS publishing; 2013
ISBN: 978-0672336973

ITAP 3362: Mobile Application Design & Development

Prerequisites

ITAP2312 – Web Programming
GEIT3341 – Database 1

Credit hours

3 Credit Hours

Course overview

The course will introduce students to the various platforms in use on small and mobile devices. Platforms will include Apple iPhone, Google Android OS, and Microsoft Windows Phone 7. Students will create applications for each platform using specialized development environments.

PMU Competencies and Learning outcomes

By the end of the course the students should be able to:

- Describe and contrast the architectures and development environments of popular mobile platforms
- Summarize and explain the development cycle for mobile apps including building, testing, and deployment
- Analyze, install and configure different tools for the development of mobile apps, e.g., Eclipse and Android Studio
- Design, develop and test apps for popular mobile platforms like Apple iOS, Google Android and MS Windows
- Justify, design and develop cross-platform web applications for mobile devices using a contemporary tool such as PhoneGap
- Demonstrate the communication, leadership and teamwork skills necessary to work in, or in charge of teams

Detailed Course Description

COEN 4393: The course will introduce students to the various platforms in use on small and mobile devices. Platforms will include Apple iPhone, Google Android OS, and Microsoft Windows Phone 7. Students will create applications for each platform using specialized development environments.

Required Text

A. Required Textbook

Thomas Duffy, Programming with Mobile Applications: Android, iOS, and Windows Phone 7. Course Technology, Cengage Learning 2013, ISBN: 9781133628132

Class Rules

Disruptive Behavior

Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance

Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness
When a student is late for 3 times it is counted as one absent.

Withdrawal
It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment
Student's performance in this course may be assessed on the basis of:
Class Participation
Quiz
Assignments
One term exams
A project
A final exam

Evaluation

Quiz	10%
Assignments	10%
Project	20%
Major Exam 1	15%
Major Exam 2	15%
Final	30%
Total	100%

Course Format

A. Instruction

This course utilizes both lecture/discussion and laboratory exercises. Students are expected to attend three hours of lecture/discussion per week and three hours of laboratory per week. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Occasionally, students have a collaborative problem solving activity.

B. Web supplement

Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:
Course syllabus
Course assignments
Sample solutions to examinations (after being graded and returned)
Sample solutions to programming exercises (after being graded and returned)
Course calendar (an active utility)
Course e-mail (an active utility)
Course discussion list (an active utility)
Student course performance (an active utility)

CLASS SCHEDULE (Tentative)

Week	Topics	Chapter Readings
Week 1	Introduction, Setup	Read Chapters, 1, Appendix A, Up For Discussion p. 15 Programming Exercises p. 15, Install necessary software
Week 2	Developing for Small Devices, Best Practices	Read Chapter 2, Up For Discussion p. 38
Week 3		Read Chapter 3, Up For Discussion p. 70 Programming Exercises p. 70
Week 4	Google Android – App Inventor	Read Chapter 4, Complete All Detective Work, Up For Discussion p. 100
Week 5		Programming Exercises p. 100
Week 6	Google Android – Android Studio	Read Chapter 5, Complete All Detective Work, Up For Discussion p. 164
Week 7		Programming Exercises P. 164
Week 8	Apple iOS	Read Chapter 6, Complete All Detective Work, Up For Discussion p. 221
Week 9		Programming Exercises p. 221
Week 10	Microsoft Windows Phone 7	Read Chapter 7, Complete All Detective Work, Up For Discussion p. 253
Week 11		Programming Exercises p. 253
Week 12	Web Applications	Read Chapter 8, Complete All Detective Work, Up For Discussion p. 289
Week 13		Programming Exercises p. 289
Week 14	PhoneGap Final Project - due	Read Chapter 9, Up For Discussion p. 312, Programming Exercises p.313

ITAP 4315: Cloud Computing

Semester Credit Hours: 3 (3, 0)

I. Course Overview

This course provides necessary background to build a cloud infrastructure based on a cloud computing reference model. The reference model includes five fundamental layers and three cross-layer functions for building a cloud infrastructure. For each layer and function, this course covers the comprising technologies, components, processes, and mechanisms. This course follows an open approach to describe concepts and technologies.

II. PMU Competencies and Learning Outcomes

This course has both theoretical and practical aspects. Students develop both the conceptual basis and the practical skills in the design and implementation of cloud-based systems to support the core business processes of an organization. They gain an understanding of and appreciation for the need for cloud technologies in large organizations for the purpose of not only planning and control but also to gain strategic competitive advantage. Additionally, this course makes an extensive use of the PMU technology infrastructure to provide communication between faculty and students. While the course does not include a structured laboratory component, assignments/projects are assigned to ensure students gain an introductory experience in working with a cloud provider such as Amazon.

III. Detailed Course Description

This course is based on "EMC Cloud Infrastructure and Services (CIS) - V2", and tailored for PMU requirements. This course educates participants on building cloud infrastructure based on a cloud computing reference model. The reference model includes five fundamental layers (physical, virtual, control, orchestration, and service) and three cross-layer functions (business continuity, security, and service management) for building a cloud infrastructure. For each layer and cross-layer function, this course covers the comprising technologies, components, processes, and mechanisms. This course takes an open-approach to describe the concepts and technologies, which are further illustrated and reinforced with EMC-related product examples. The course follows the U.S. NIST as a guide for all definitions of cloud computing. Upon completing this course, participants will have the knowledge to make informed decisions on technologies, processes, and mechanisms required to build a cloud infrastructure.

IV. Requirements Fulfilled

This course satisfies three hours of the requirements for degrees in Information Technology. It is an available elective for the degree in Computer Science and Computer Engineering. It should be taken no earlier than the junior year.

V. Required Prerequisites

ITAP 2431 - Network Management
GEIT 3331 - Computer Organization

VI. Learning Outcomes
In this course, students learn:
CLO1 To describe cloud computing concepts, deployment models, and service models
CLO2 To recognize the reference model and the key considerations to build a cloud infrastructure
CLO3 To setup and configure the key components and processes required to build the physical,

virtual, control, orchestration, and service layers of a cloud infrastructure

CLO4 To analyze and manage the business continuity, security, and service management functions of a cloud infrastructure

CLO5 To deploy and integrate software application over cloud infrastructure

CLO6 To develop improved communication and collaborative skills

VII. Assessment Strategy

Course grades are be based on

- Regularly assigned homework to motivate students to do the work and earn credit accordingly.
- Weekly structured in-class exercises designed to guide students through specific course topics.
- Several in-class quizzes/exams to assess students' accumulative mastery of content covered prior to time of exam.
- A comprehensive final exam to assess students' accumulative mastery of course material.

Students' final grades are based on 30% credit for homework and projects, 5% for participation in classroom discussion, 25% on major exams, 15% on quizzes and 25% for the final examination.

Students are required to maintain a journal of thoughts and commentaries during the course. The journal contains daily entries including the identification of areas of interest and concern, notes on the preparation of presentation and comments and analysis of classmate's presentations. The journal is reviewed weekly by the instructor to provide feedback to the students.

Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he proceeds through the curriculum.

VIII. Course Format

This course utilizes both lecture/discussion and in-class exercises. Students are expected to attend three hours of lecture/discussion per week. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Once a week students should have at least 30 minutes of collaborative problem solving activity.

Classroom Hours (6 hours per week) Class: 3

Lab: 0

Web supplement: Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:
Course syllabus
Course assignments
Sample solutions to examinations (after being graded and returned)
Course calendar (an active utility)
Course e-mail (an active utility)
Course discussion list (an active utility)
Student course performance (an active utility)

IX. Topics to be covered

Module 1: Introduction to Cloud Computing
- Essential characteristics of cloud computing

- Cloud service models and cloud service brokerage
- Cloud deployment models

Module 2: Building the Cloud Infrastructure
- Cloud computing reference model
- Deployment options and solutions for building cloud infrastructure
- Considerations for building cloud infrastructure

Module 3: Physical Layer
- Compute system
- Storage system architectures
- Network connectivity

Module 4: Virtual Layer
- Virtual layer functions
- Virtualization software
- Resource pool and virtual resources

Module 5: Control Layer
- Control layer functions
- Control software
- Software-defined approach for managing IT infrastructure
- Resource optimization techniques

Module 6: Service and Orchestration Layers
- Service layer functions
- Cloud portal
- Cloud interface standards
- Protocols for accessing cloud services
- Service orchestration
- Cloud service lifecycle

Module 7: Business Continuity
- Business continuity and service availability
- Fault tolerance mechanisms
- Backup and replication
- Cloud application resiliency

Module 8: Security
- Cloud security threats
- Cloud security mechanisms
- Governance, risk, and compliance

Module 9: Service Management
- Service portfolio management processes
- Service operation management processes

X. Laboratory Exercises

N/A

XI. Technology Component

This course makes use of the university's wireless access infrastructure during the class/lecture sessions. The course relies on the university and the students having access to an isolated professional grade network environment for practical exercises and demonstrations.

XII. Special Projects/Activities

Basic concepts of Cloud Computing in terms of Application Development, Deployment and Management
Cloud setup architecture designing based on the application requirements
Amazon Web Services basics (Background/Pricing/Regions/OS) and overview of other cloud service providers
Setup of the AWS EC2 Compute Instance, AMI- OS Image Management
Setup of S3 Storage, CloudFront, Elastic Transcoder, RDS and SES Cloud Services
Setup of VPC, Route53, Security Management (IAM, Security Groups, Polices, SNS)
Application Deployment and Integration over Cloud Infrastructure

XIII. Textbooks and Teaching Aids

A. Required Textbook

Main Textbook: Cloud Computing: Concepts, Technology & Architecture by Thomas Erl, Ricardo Putini & Zaigham Mahmood by Prentice Hall, 1st Ed. 2013. ISBN: 978-0133387520

B. Alternative Textbooks

Alternative Textbook: Cloud Computing Explained: Implementation Handbook for Enterprises by John Rhoton by Recursive Press, 2nd Ed. 2009 ISBN: 978-0956355607

C. Supplemental Print Materials

EMC CIS v2 presentations and posters

D. Supplemental Online Materials

As available from publisher

ITAP 4361: Operating Systems

Semester Credit Hours: 3 (3,0)

I. Course Overview

This course is the study of the principles, purposes, and organization of operating systems. The goal is to prepare students an understanding of the theory as well as practices of the design and implementation of operating systems software.

II. PMU Competencies and Learning Outcomes

Students in this course develop conceptual and programming skills necessary for continued success in computer science. The skills enhance their abilities to appreciate the theory and practices of operating systems common to computer science as a discipline and to effectively communicate their solutions to fellow professionals. This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course includes individual as well as group projects, establishes both conceptual reasoning skills and technical communication skills, and provides opportunities for the presentation and defense of designed solutions.

III. Detailed Course Description

COSC 4361: The Operating Systems course is concerned with the study of the principles, purposes, and organization of operating systems, including processes, tasks, scheduling, inter-process communication, synchronization, mutual exclusion, memory management, device management, file systems, security and protection, multi-CPU systems, computer networking, and distributed computing.
This course satisfies three hours of the requirements for the degree in computer science. It is required of all students pursuing a degree program in computer science within the College of Computer Engineering and Science. It should be taken in the first semester of the senior year.

IV. Requirements Fulfilled

This course is required for all students majoring in Computer Science and Computer Engineering in the College of Computer Engineering and Science. It is also recommended as an elective for students majoring in IT.

V. Required Prerequisites

GEIT 1412: Computer Science II
GEIT 2421: Data Structure
GEIT 3331: Computer Organization
COSC 3411: Systems Programming

VI. Learning Outcomes

In this course, students will be able to:

1. Describe the underlying structure and services provided by operating systems
2. Describe and differentiate between the concepts of threads and processes
3. Explain and evaluate merits of different process scheduling algorithms
4. Explain inter process communication and process synchronization issues in cooperating processes or threads.
5. Explain the structure and design of memory and storage management in computer systems.

6. Develop the communication, leadership and teamwork skills necessary to work in, or in charge of teams.

VII. Assessment Strategy

This course is designed with three primary goals in mind: to introduce students to the conceptual basis and practical issues associated with the compiler construction, to enhance the student's programming techniques to its application in computer science, and to provide students with the opportunity to communicate their designs and implementations to their peers in a professional setting. With this in mind, the course grade involves an assessment of their performance on examinations that focus on the application of programming paradigms to the solutions of problems, the performance analysis of the designed solutions, and the communication of designed solutions to those problems to an audience. Course grades are based on:
Homework to motivate students to do the work and earn credit accordingly.
Announced and unannounced quizzes.
Midterm exam.
A comprehensive final exam.

The final grade is based on 20% credit for the quizzes, 25% on programming assignments, 20% for in-class midterm exam and 30% for the final examination. 5% marks will be awarded based on class participation. Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he/she proceeds through the curriculum.

Evaluation

Class Participation	5 %
Programming Assignments	25 %
Quizzes	20%
Midterm Exam	20 %
Final Exam	30 %
Total 100%	

Grading Scale

A+	96 - 100%
A	90 - 95 %
B+	86 – 89 %
B	80 – 85 %
C+	76 – 79 %
C	70 – 75 %
D+	66 – 69 %
D	60 – 65 %
F	Below 60 %

VIII. Course Format

This course utilizes a mix of in-class lectures, discussions, and software demonstrations designed to help students learn the various tasks involved in designing and implementation of an operating system. While class meetings are utilized to emphasize conceptual foundation in topics related to these tasks, software demonstrations are used to provide students with hands-on training in performing these tasks. Students are expected to attend three hours of lecture per week. There are no scheduled lab hours for this course.

In addition, the instructor should consider creating a Web site for this course using Web technologies such as WebCT or BLACKBOARD. At minimum, the site should include:
Course syllabus
Lecture material (for example PowerPoint slides, lecture notes, etc.). These should be placed on the

site ahead of class meeting so that students may use the material to prepare for the lecture
Software demonstration exercises completed in class
Out-of-Class assignments and projects
Mechanism for students to digitally submit their assignments
Course calendar
Mechanism to communicate electronically (for example e-mail)
Discussion groups
Students course performance measures
Classroom Hours (3 hours per week) Class: 3

Lab: 0

IX. Topics to be covered

Week	Chapters	Topics
1	Introduction/ Chapter 1	Introduction
2	Chapter1/Chapter 3	Introduction (continued), Virtual Machines (Vmware)
3	Chapter 3	Process Concept
4	Chapter 4	Multithread Programming
5	Chapter 5	Process Scheduling
6-7	Chapter 6	Synchronization
8	Chapter 7	Deadlock
9-10	Chapter 8	Memory Management/ MidTerm Exam
11-12	Chapter 9	Virtual Memory
13-14	Chapter 10, Chapter 11	File Systems
15	Chapter 2 & 12	Systems Structures /Secondary Storage Structure
16	Final Exam	(Final exam. Scheduled by the Registrar)

X. Laboratory Exercises

There is no lab component for this course. However, the students are assigned one out-of-class application development exercise every three weeks. These hands-on exercises are designed to illustrate various development concepts covered during lecture meetings. Thus, students are expected to complete about four such exercises. Collectively, these exercises enable students to learn the details of the operating systems. The following major areas should be covered in these exercises:
Process and thread management
Process synchronization
Memory management
File system management

XI. Technology Component

A. In class, the instructor makes use of state-of-the art multimedia projection equipment and

software. These are used to project slides and Web-based content relevant to the concepts of operating systems.

B. Outside class, the instructor uses Web-based course management software (for example WebCT, BLACKBOARD) to interact with students as described under course format section.

XII. Special Projects/Activities

There will be multiple programming based mini-projects in this course. These projects are designed to permit students to apply concepts, methods, and tools learned in class. The students are expected to:
Use operating system calls to create and manage processes
Use operating system calls to create and manage thread
Use system calls for memory and file system management

XIII. Textbooks and Teaching Aids

A. Required Textbook

Silberschatz, A., Operating Systems Concepts. 9th Edition, Wiley, 2012.
ISBN: 978-8126554270

B. Alternative Textbooks

Tanenbaum, A., Modern Operating Systems, 4th Edition, Prentice Hall, 2014.
ISBN 978-1292061429.

C. Supplemental Print Materials

As available from publisher.

D. Supplemental Online Materials

As available from publisher.

ITAP 4390: Information Security and Assurance

Semester Credit Hours: 3 (3,0,3)

I. Course Overview

The purpose of this course is to provide an overview of methods to assure secure and confidential information systems. Coverage includes basic concepts of main security and privacy issues of the Internet, and devices and implementation of security methods for computer networks and the Internet.

II. PMU Competencies and Learning Outcomes

This course helps students understand the critical issues involved with securing corporate information from accidental destruction and malicious attack. Additionally, the course provides students with the communication, leadership and teamwork skills necessary to effectively work as professionals in teams, or in charge of teams, responsible for security projects.

III. Detailed Course Description

This course examines the nature of threats to corporate computer systems and the systems, polices, and protocols necessary to safeguard data and mitigate both accidental and deliberate threats. The course considers the legal, managerial, and professional aspects of threat mitigation, together with in-depth coverage of specific areas of concern including single and multi-factor authentication, authentication and role-based access control systems, intrusion detection, the identification of critical assets, and the use of cryptography to secure communication.

IV. Requirements Fulfilled

This course is required of all students majoring in Information Technology in the College of Computer Engineering and Science.

V. Required Prerequisites

ITAP 3431 Network Security

VI. Learning Outcomes
Upon completion of this course, students will be able to:

- Identify the threats posed to information security and the more common attacks associated with those threats.
- Analyze the legal, ethical, managerial and professional aspects of information security.
- Identify e-mail and web services vulnerabilities and demonstrate how to safeguard against them.
- Identify and apply the various approaches to authentication and remote access protection.
- Identify and describe the categories and operating models of networking security architectures and intrusion detection systems.
- Discuss various approaches to access control, including use of biometric access mechanisms.

Apply the basic principles of cryptography.

VII. Assessment Strategy

This course is designed with three primary goals in mind: to further the student's understanding of data security issues, to lead students to connect those issues with best business practices in defending corporate data from internal and external threats, and to encourage the students to communicate their ideas and their expertise to the professional community. With this in mind, the course grade involves an assessment of their performance on in-class quizzes and exams that focus on the application of best practices to resolving critical data security problems.

Weekly assigned homework to motivate students to do the work and earn credit accordingly.

Weekly in-class quizzes.

Weekly, in-class presentations by students of solutions to real world problems related to the course material and classroom discussion and critique of the presentation.

Three in-class exams to assess the student's accumulative mastery of content covered prior to time of exam.
A comprehensive final exam to assess the student's accumulative mastery of course material.

The final grades is based on 15% credit for the homework, 15% for the quizzes, 10% for the presentations and participation in classroom discussion, 30% on in-class exams, and 30% for the final exam.

Students are required to maintain a journal of thoughts and commentaries during the course. The journal contains daily entries including the identification of areas of interest and concern, notes on the preparation of presentation and comments and analysis of classmate's presentations. The journal is reviewed weekly by the instructor to provide feedback to the students.

Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he proceeds through the curriculum.

VIII. Course Format

A. Instruction
Primary instruction is a lecture format, with the course meeting three times per week for one hour each meeting. At least once per week students should be prepared to make presentation on a topic selected by the instructor and to take part in a discussion based on that presentation. Once a week, students should have at least 30 minutes of collaborative problem-solving activity.

B. Web supplement
Course home page on the University's BLACKBOARD system should contain the following:
Course syllabus
Course assignments
Keys to quizzes and exams (after students have completed them)
Course calendar (an active utility)
Course e-mail (an active utility)
Course discussion list (an active utility)
Students course marks (an active utility)
Classroom Hours (3 hours per week) Class: 3

Lab: 0

IX. Topics to be Covered

Introduction to Information Security
The need for security
Legal, ethical and professional issues in Information Security
CSI/FBI computer and security survey
Security technology: Firewalls and VPN
Planning for continuity
Cryptography
Physical security
Information security maintenance
Information Systems Security: Cases of Network Administrator Threat
Presentation

X. Laboratory Exercises

This course does not require laboratory exercises.

XI. Technology Component

This course has no technology component other than use of the student's personal laptop computers as appropriate.

XII. Special Projects/Activities

Students are required to keep a "reflective notebook" in which, after each class, they enter their own assessments of what they learned, and what questions remain from the class. From each exercise set, each student selects one problem, which the student thinks best reflects the way the topic is used in a technical context. A detailed solution to the problem is included in the student's reflective notebook.

XIII. Textbooks and Teaching Aids

A. Required Textbook

Michael E. Whitman and Herbert J. Mattord, Principles of Information Security, 6th Edition, 2018, Cengage
ISBN-13: 9781337578738

B. Alternative Textbook

Andress, J., The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice, Syngress, 2011.
ISBN 978-1597496537

C. Supplemental Print Materials

As available from publisher.

D. Supplemental Online Materials

As available from publisher.

E. COMPUTER SCIENCE COURSES

- COSC 2312: Web Programming
- COSC 3332: Discrete Structures and Combinatorial Analysis
- COSC 3361: Computer Networks
- COSC 3351: Algorithms
- COSC 3411: Systems Programming
- COSC 4361: Operating Systems
- COSC 4461: Programming Languages
- COSC 4362: Artificial Intelligence
- COSC 4363: Theory of Computation

CS Elective courses:

- COSC 3354: Introduction to Cryptography
- COSC 4371: Computer Graphics
- COSC 4374: Computer Vision
- COSC 4393: Special Topics
- COSC 4311: Parallel Computing
- ITAP 3313: User interface Development
- ITAP 4371: E-commerce
- ITAP 3371: Database II

- COSC 3359: Computer Animation
- COSC 3357: Logic and Formal Verification
- COSC 4352: Formal Methods in Software Engineering
- COSC 4372: Distributed Systems and Algorithms
- COSC 4376: Bioinformatics
- COSC 4380: Quantum Computing and Computation
- COSC 4364: Compilers

COSC 2312: Web Programming

Prerequisites
GEIT 1411: Computer Science I

Credit hours
4 Credit Hours

Course overview

This course is designed to provide the students with an introduction to World Wide Web programming. It introduces the student to the techniques used in programming web pages for interactive content. It specifically addresses the basic elements of AJAX -Asynchronous JavaScript and XML to design web pages that dynamically interact with databases that reside on a server. The course begins by reviewing basic web technologies such as HTML, XHTML, CSS style sheets, and explores the use of event-driven programming in JavaScript to add interactive elements such as buttons and text fields to web pages. Students will be able to use AJAX tools to build web pages that connect to servers like Google to dynamically access data (maps, search results, videos, images, etc).

PMU Competencies and Learning outcomes

By the end of the course the students should be able to:
Construct a web page using the latest HTML technology
Design a web page with the suitable formatting technology e.g., CSS3
Apply event-driven programming in web application development
Develop and implement a web application using JavaScript framework (for example JQuery)
Develop the communication, leadership and teamwork skills necessary to work in or lead teams

Detailed Course Description

ITAP 1312: The course begins by reviewing basic web technologies such as HTML, XHTML, CSS style sheets, and explores the use of event-driven programming in JavaScript to add interactive elements such as buttons and text fields to web pages. Students will be able to use AJAX tools to build web pages that connect to servers like Google to dynamically access data (maps, search results, videos, images, etc).

Required Text

A. Required Textbook

Deitel, J. & Deitel, M. (2009). Internet & World Wide Web How To Program; 5tht edition. Pearson Education. ISBN 0-13-603542-6

Class Rules

Disruptive Behavior
Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance
Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness

When a student is late for 3 times it is counted as one absent.

Withdrawal

It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment

Student's performance in this course may be assessed on the basis of:
Homework
Two term exams
A programming project
Lab exercises
A final exam

Evaluation

Homework Assignments	5%
Programming Project	15%
Lab	20%
Exam 1	15%
Exam 2	15%
Final	30%
Total	100%

Course Format

A. Instruction

This course utilizes both lecture/discussion and laboratory exercises. Students are expected to attend three hours of lecture/discussion per week and three hours of laboratory per week. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Occasionally, students have a collaborative problem solving activity.

B. Web Supplement

Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:
Course syllabus
Course assignments
Sample solutions to examinations (after being graded and returned)
Sample solutions to programming exercises (after being graded and returned)
Course calendar (an active utility)
Course e-mail (an active utility)
Course discussion list (an active utility)
Student course performance (an active utility)

CLASS SCHEDULE (Tentative)

Week#	Lecture (3 hours/week)	Lab (2hours/week)	Projects/ Readings
Week #1	Syllabus Introduction to Computers and the Internet	Laboratory exercises	Read Chapter 1
Week #2	Web Browser Basics: Internet Explorer and Firefox Dive into® Web 2.0	Laboratory exercises	Read Chapter 2 Read Chapter 3
Week #3	Dive into® Web 2.0 HTML Basics	Laboratory exercises	Read Chapter 3
Week #4	Introduction to XHTML	Laboratory exercises	Read Chapter 4
Week #5	Introduction to XHTML Cascading Style Sheets™ (CSS)	Laboratory exercises	Read Chapter 4 Read Chapter 5
Week #6	Cascading Style Sheets™ (CSS)	Laboratory exercises	Read Chapter 5
Week #7	JavaScript: Introduction to Scripting	Laboratory exercises	Read Chapter 6
Week #8	JavaScript: Introduction to Scripting	Laboratory exercises	Read Chapter 6
Week #9	JavaScript: Control Statements I	Laboratory exercises	Read Chapter 7
Week #10	JavaScript: Control Statements II	Laboratory exercises	Read Chapter 8
Week #11	JavaScript: Functions	Laboratory exercises	Read Chapter 9
Week #12	JavaScript: Arrays	Laboratory exercises	Read Chapter 10
Week #13	JavaScript: Objects	Laboratory exercises	Read Chapter11
Week #14	Document Object Model (DOM): Objects and Collections	Laboratory exercises	Read Chapter 12 Group Project presentations
Week #15	JavaScript: Events	Laboratory exercises	Read Chapter 13 Group Project presentations

COSC 3332: Discrete Structures and Combinatorial Analysis

Prerequisites

GEIT 2331: Math. Reasoning & Algorithmic Thinking

Credit hours

3 Credit Hours

Course overview

Discrete Structures and Combinatorial Analysis is the study of objects that have discrete as opposed to continuous values including counting techniques, relations, graphs, trees and combinatorics.

PMU Competencies and Learning outcomes

Students of COSC 3332: Discrete Structures and Combinatorial Analysis develop the quantitative skills necessary for continued success in computer science. These skills enhance their ability to both analyze and describe mathematically many of the algorithms and data structure performance characteristics common to computer science as a discipline and to effectively communicate their solutions to fellow professionals. The course makes extensive use of the technology infrastructure of the school for communication within the class and between students and the instructor. Through the use of group tasks and projects this course establishes both mathematical reasoning skills and technical communication skills.

Detailed Course Description

Together with GEIT 2331: Mathematical Reasoning and Algorithmic Thinking, this course develops the quantitative skills necessary for continued success in computer science. COSC 3332: Discrete Structures and Combinatorial Analysis is concerned with the application of objects with discrete characteristics to computer science as a discipline in order that commonly used structures may be described, characterized and analyzed. It focuses on the study of objects that have discrete as opposed to continuous values including counting techniques, relations and combinatorics. Graphs, including Euler and Hamilton paths and shortest path problems are examined, as are tree applications.

Learning Outcomes

In this course, students learn:

- To understand and apply basic and advanced counting techniques.
- To use recurrence relations in order to analyze divide-and-conquer algorithms.
- To familiarize themselves with representing, characterizing and manipulating relations in order to model discrete objects and their relationships.
- To develop an understanding of how graph and tree concepts are used to solve problems arising in computer science.
- To communicate the solutions of technical problems to other professionals.

Required Text

A. Required Textbook

Rosen, K. H. (2013) Discrete Mathematics and Its Applications. 7th Edition, McGraw-Hill.
ISBN 978-0-07-131501-2

B. Alternative Textbooks

Richard Johnsonbaugh. (2005) Discrete Mathematics,
Sixth Edition, Prentice Hall.
ISBN 0-13-117686-2

Hall, C., & O'Donnell J. (2000) Discrete Mathematics Using a Computer, Springer Verlag
ISBN 1-85-233089-9

Balakrishnan, V.K. (1996). Introductory Discrete Mathematics, Dover
ISBN 0-48-669115-2

Class Rules

- Disruptive Behavior
  - Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
  - Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

- Class attendance

Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness

When a student is late for 3 times it is counted as one absent.

Withdrawal

It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment

Grades shall be computed on the basis of the following:

Class participation	5%
Homeworks	15%
Quizzes	20%
Exam I	15%
Exam II	15%
Final Exam	30%
Total	**100%**

Course Format

A. Instruction

Primary instruction is a lecture format, with the course meeting for three hours per week.

B. Web supplement

Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:

- Course syllabus
- Lecture material (PowerPoint slides, lecture notes, etc.). These will be placed on the site ahead of class meeting so that students may use the material to prepare for the lecture.
- Course assignments
- Sample solutions to examinations (after being graded and returned)

CLASS SCHEDULE (Tentative)

Week#	Topics covered	Sections to Read	Practice Problems
1	Basics of Counting The Pigeonhole Principle	6.1 6.2	6, 12, 17, 21, 30, 34, 38. 10, 18, 26, 30, 32.
2	Permutations and Combinations Binomial Coefficients and Identities	6.3 6.4	8, 12, 16, 22, 26. 4, 8, 12, 16, 20, 23.
3	Generalized Permutations and Combinations	6.5	6, 14, 18, 23, 30, 40.
4	Recurrence Relations Solving Linear Recurrence Relations	8.1 8.2	10, 20, 22, 24. 2, 10, 16, 22, 26, 34.
5	Divide-and-Conquer Algorithms and Recurrence Relations	8.3	4, 8, 10, 16, 18, 28.
6	Inclusion-Exclusion Principle Applications of Inclusion-Exclusion	8.5 8.6	4, 8, 10, 14, 18. 4, 6, 14, 20.
7	Relations and Their Properties n-ary Relations and Their Applications	9.1 9.2	
8	Representing Relations Closures of Relations	9.3 9.4	
9	Equivalence Relations Partial Orderings	9.5 9.6	
10	Graphs and Graph Models Graph Terminology and Special Types of Graphs	10.1 10.2	
11	Representing Graphs and Graph Isomorphism Connectivity	10.3 10.4	
12	Euler and Hamilton Paths	10.5	
13	Trees Tree Traversal	11.1-3	
14	Minimum Spanning Trees	11.4-5	

COSC 3361 : Computer Networks
COEN 3361: Computer Networks

Semester Credit Hours: 3 (3-0-3)

I. Course Overview

This course covers the architecture and protocols of local and wide area networks, including signaling, data representation, error control, flow control and routing. The C++ or Java-based sockets API is used to provide practical examples of communication, error control and flow control. Peer to peer and client/server configurations based upon Unix/Linux and Windows architectures are explored.

II. PMU Competencies and Learning Outcomes

This course concentrates on theoretical and technical issues. Students in this course enhance their interpersonal and group effectiveness skills.

III. Detailed Course Description

COEN 3361: Computer Networks is concerned with the structure of data communications; from the electric interface, flow control, medium access control protocols, through data transmission and network protocols, packet switching and frame relay protocols, and includes an examination of network standards and open systems.

IV. Requirements Fulfilled

This course is a required course for students in the Computer Engineering and Computer Science programs in the College of Computer Engineering and Science (CCES). It should be taken no earlier than the senior year.

V. Required Prerequisites

MATH 2313: Probability and Statistical Methods

MATH 3433: Linear Algebra and Differential Equations

GEIT 3421: Data Structures

VI. Learning Outcomes

In this course, students learn:

- Develop an understanding of the principles of electronic communication (physical layer) and the practical application to computer networks.
- Explain the concepts, issues and protocols related to data link layer
- Explain the concepts, issues and protocols related to network layer
- Enlist the issues and protocols related to transportation and application layers
- Develop improved communication and collaborative skills.

VII. Assessment Strategy

This course is designed to introduce students to the physical, theoretical and practical principles underlying computer networks. Course grades are be based on:

- Regular quizzes and home works to motivate students to refresh their concepts
- Presentation based on their readings in current network literature.
- Two in-class exams to assess the student's accumulative mastery of content covered prior to

time of exam.
- A comprehensive final exam to assess the student's accumulative mastery of course material.

The final grades is based on 25% credit for the homework, participation & presentations, 20% for the quizzes and academic writing assignments, 25% on in-class exams and 30% for the final examination.

VIII. Course Format

This course is primarily a lecture course. Students are expected to attend three hours of lecture per week.

Classroom Hours (3 hours per week) Class: 3

Web supplement: Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:

- Course syllabus.
- Course assignments.
- Keys to exams (after students have completed them).
- Course calendar (an active utility).
- Course e-mail (an active utility).
- Course discussion list (an active utility).
- Students course marks. (an active utility).

IX. Topics to be covered

This course follows Tanenbaum 5e textbook format. The broad topics that will be covered are:

A. Introduction to data communications, reference models, networking technologies
B. Physical layer (electrical interface, transmission media, channel capacity)
C. Data link layer (data representations, Error control/ flow control)
D. Media Access Control
E. Network layer (addressing, routing, congestion control)
F. Transport layer (TCP, UDP, congestion control)
H. Application protocols (SMTP, HTTP, etc)

X. Laboratory Exercises

There are no lab exercises for this course.

XI. Technology Component

This course makes use of the university's wireless access infrastructure during the class/ lecture sessions. In addition, the course makes use of the university's Blackboard interactive communication tool to enhance communication between the instructor and the students.

XII. Special Projects/Activities

There are four programming activities that are designed to develop a practical understanding of key data communication issues:

- Packet tracer exercises related to router, switch, and VLAN configurations.
- Client server application development using socket programming
- Wireshark practical exercises

XIII. Textbooks and Teaching Aids

A. Required Textbook
Tanenbaum, Andrew S., Computer Networks, 5th Edition, Prentice Hall, 2014.
ISBN 978-0132126953

B. Alternative Textbooks

Stallings, William, Data and Computer Communications, 10th Edition, Prentice Hall, 2014.

C. Supplemental Print Materials

As available from publisher.

D. Supplemental Online Materials

As available from publisher

COSC 3351: Algorithms

Semester Credit Hours: 3 (3,0)

I. Course Overview

This course is the study of the design and performance analysis of algorithms. Time and space complexity analysis of algorithms, design paradigms, and graph algorithms are discussed.

II. PMU Competencies and Learning Outcomes

Students of COSC 3351: Algorithms develop quantitative and programming skills necessary for continued success in computer science. The skills enhance their abilities to devise, analyze, and comprehend mathematically the performance characteristics of algorithms and various design paradigms common to computer science as a discipline and to effectively communicate their solutions to fellow professionals. This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course includes individual as well as group projects, establishes both mathematical reasoning skills and technical communication skills, and provides opportunities for the presentation and defense of designed solutions.

III. Detailed Course Description

This course is concerned with the study of the design and performance analysis of algorithms, including maximum contiguous subarray, divide-and-conquer algorithms, graph algorithms, dynamic programming, and greedy algorithms. One important lasting effect of this course is to enhance and develop the ability to specify, design, implement, test, and analyze solutions to programming problems utilizing the proven design paradigms presented in this course.

IV. Requirements Fulfilled

COSC 3351: Algorithms satisfies 3 hours of the requirements for the degree in computer science. It is required of all students pursuing a degree program in computer science within the College of Computer Engineering and Science. It should be taken in the second semester of the junior year.

V. Required Prerequisites

- COSC 3421: Data Structures
- MATH 1313: Statistical Methods

VI. Learning Outcomes

In this course, students learn:

1. To understand and apply performance analysis techniques to analyze sorting algorithms.
2. To understand and analyze the performance of divide-and-conquer algorithms.
3. To understand and analyze the performance of graph algorithms.
4. To understand and analyze the performance of dynamic programming algorithms.
5. To understand and analyze the performance of greedy algorithms.
6. To understand the concepts of complexity classes.
7. To use mathematical and measurement techniques to analyze the performance characteristics of algorithms.
8. To develop improved communication and collaborative skills.

VII. Assessment Strategy

This course is designed with three primary goals in mind: to introduce students to the conceptual basis and practical issues associated with the use, development, and analysis of designed algorithms, to lead students to connect the mathematics to its application in computer science, and to provide students with the opportunity to communicate their designs and implementations to their peers in a professional setting. With this in mind, the course grade involves an assessment of their performance on examinations that focus on the application of programming paradigms to the solutions of problems, the performance analysis of the designed solutions, and the communication of designed solutions to those problems to an audience. Course grades are based on:

- Weekly assigned homework to motivate students to do the work and earn credit accordingly.
- Weekly, in-class presentations by students of solutions to real world problems related to the course material and classroom discussion and critique of the presentation.
- Two in-class examinations to assess the student's accumulative mastery of content covered prior to the time of the examination.
- Three programming assignments testing students understanding of the major concepts introduced during the course.
- A comprehensive final examination to assess the student's accumulative mastery of course material.

Class participation	5%
Homeworks	15%
Quizzes	15%
Project	10%
Midterm Exam	25%
Final Exam	30%
Total	**100%**

Students are required to maintain a journal of thoughts and commentaries during the course. The journal contains daily entries including the identification of areas of interest and concern, notes on the preparation of presentation and comments and analysis of classmate's presentations. The journal is reviewed weekly by the instructor to provide feedback to the students.

Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he proceeds through the curriculum.

VIII. Course Format

A. Instruction

This course utilizes both lecture/discussion and laboratory exercises. Students are expected to attend three hours of lecture per week. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Once a week students should have at least 30 minutes of collaborative problem solving activity.

B. Web supplement

Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:

- Course syllabus
- Course assignments
- Sample solutions to examinations (after being graded and returned)
- Sample solutions to programming assignments (after being graded and returned)
- Course calendar (an active utility)
- Course e-mail (an active utility)
- Course discussion list (an active utility)
- Student course performance (an active utility)

IX. Topics to be Covered

A. Mathematical foundations to analyze complexity of algorithms
   1. big Oh, big Omega, big Theta notations.
   2. Solving recurrences.
B. Divide-and-conquer algorithms
   1. Merge sort
   2. Polynomial multiplication
   3. Randomized selection
C. Graphs
   1. Notations
   2. Breadth-first search and depth-first search
   3. Cycle finding and topological sort
   4. Maximum spanning trees
   5. Dijkstra's shortest path algorithm
D. Dynamic programming
   1. Knapsack
   2. Chain matrix multiplication
   3. Longest common subsequence
   4. All pairs shortest path
E. Greedy algorithms
   1. Activity selection
   2. Huffman coding
F. Complexity classes
   1. Nondeterminism
   2. Classes P and NP
   3. NP-complete problems
   4. Polynomial reductions

X. Laboratory Exercises

This course does not offer a separate laboratory to students.

XI. Technology Component

This course makes use of the university's wireless access infrastructure. The course relies on the university and the students having access to professional grade application development environments for the students to use.

XII. Special Projects/Activities

Students are required to keep a "reflective notebook" in which, after each class, they enter their own assessments of what they learned, and what questions remain from the class. From each exercise set, each student selects one problem, which the student thinks best reflects the way the topic is used in a technical context. A detailed solution to the problem is included in the student's reflective notebook.

XIII. Textbooks and Teaching Aids

A. Required Textbook

Cormen, T., C., R. Rivest Leiserson, and C. Stein. Introduction to Algorithms. _____: The MIT Press, 2001.
ISBN 0-262-03293-7

B. Alternative Textbooks

1. Bentley, J. Programming Pearls. _____: Addison-Wesley, 2000.
   ISBN 0-201-65788-0
2. Garey, M.R., and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, 1979.
   ISBN 0-7167-1045-5

C. Supplemental Print Materials

None.

D. Supplemental Online Materials

As available from publishers.

COSC 3411: Systems Programming

Semester Credit Hours: 4 (3,1)

I. Course Overview

Systems programming is the study of the basic programming principles and skills for building systems software, including the introduction to UNIX, shell programming, C, and Python programming.

II. PMU Competencies and Learning Outcome

Students in this course develop skills necessary for building systems software over UNIX platform. These skills are necessary for continued success in computer science. This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course includes a structured laboratory component to ensure that students gain the necessary experience and skill in managing the concepts introduced in the class. The course includes individual as well as group projects and provides opportunities for the presentation and defense of designed solutions.

III. Detailed Course Description

COSC 3411: Systems Programming is concerned with the basic programming principles and skills for building systems software, including the introduction to UNIX, shell programming, C, and Python programming. The course presents the students with the concepts of UNIX editor, utilities, file systems, links and shells, shell programming, C, and Python programming. Students are exposed to a variety of techniques for the implementation and uses of systems software. One important lasting effect of this course is to enhance and develop the ability to specify, design, implement and test solutions to programming problems utilizing the data structures and proven algorithms presented in this course.

IV. Requirements Fulfilled

This course satisfies four hours of the requirements for the degree in computer science. It is required of all students pursuing a degree program in computer science within the College of Computer Engineering and Science. It should be taken in the first semester of the junior year.

V. Required Prerequisites

GEIT 1412: Computer Science II
GEIT 1311: Computer Organization

VI. Learning Outcomes

In this course, students learn:
Describe the Unix environment and applications.
List and apply Unix commands used in systems administration for file, users, and process management.
Write Shell and Python scripts for file manipulation and task automation.
Write C programs in the in the context of systems programming
Develop improved communication and collaborative skills

VII. Assessment Strategy

This course is designed with three primary goals in mind: to introduce students to the conceptual basis and practical issues associated with the use and development of systems software, to provide students with significant experience in the development of systems software within a profession development environment, and to provide students with the opportunity to communicate their designs and implementations to their peers in a professional setting. With this in mind, the course grade involves an assessment of their performance on examinations that focus on the application of programming techniques to the solutions of problems and the communication of designed solutions to those problems to an audience. Course grades are based on:

- Weekly assigned homework to motivate students to do the work and earn credit accordingly.
- Weekly structured laboratory exercises designed to guide students through specific course topics.

- Midterm examinations to assess the student's accumulative mastery of content covered prior to the time of the examination.
- Five programming assignments testing students understanding of the major concepts introduced during the course.
- A comprehensive final examination to assess the student's accumulative mastery of course material.

The final grade is based on 25% credit for the programming assignments and project, 5% for the participation in classroom discussions, 20% for weekly laboratory exercises, 20% for midterm examinations, 10% for quizzes, and 20% for the final examination.

Students are required to maintain a journal of thoughts and commentaries during the course. The journal contains daily entries including the identification of areas of interest and concern, notes on the preparation of presentation and comments and analysis of classmate's presentations. The journal is reviewed weekly by the instructor to provide feedback to the students.

Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he proceeds through the curriculum.

Assessment:

Assignments/Projects		10%
Quizzes	15%	
Midterm/Majors	25%	
Final exam	25%	
Lab Work	20%	
Participation		5%

Grading Scale: Standard

A+	96 - 100%
A	90 - 95 %
B+	86 – 89 %
B	80 – 85 %
C+	76 – 79 %
C	70 – 75 %
D+	66 – 69 %
D	60 – 65 %
F	Below 60 %

Class Rules:

Disruptive Behavior
Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.
Eating / Drinking / Smoking: students are requested to refrain from engaging in these activities while in class.

Class attendance
Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to

the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness
If a student is late for 3 times it is counted as one absence.

Make-up
Midterm & Final exams – different from and harder than in-class exams. You should a valid excuse as per PMU policies

Late submissions – score reduced by 10% points per day after due date. No work will be accepted after one week of the due date.

Withdrawal
It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Academic Honesty and Integrity
The instructor strictly adheres to all university policies regarding academic integrity.
Academic dishonesty includes but not limited to the following:
Cheating on examination or other academic work,
Plagiarism and
Collusion which means unauthorized collaboration with another in preparing work offered for academic credit.
Academic dishonesty will not be tolerated and the PMU academic regulations will be strictly applied. Unless specifically expressed by the instructor, collaboration between students in this course, between students in previous courses, external assistance in any form or presenting resources/research without proper citation which has been developed by another individual or organization is strictly prohibited. ALL WORK MUST BE THE RESULT OF YOUR OWN EFFORTS.

Miscellaneous
Mobile phones: A student whose mobile rings during class will be asked to leave the classroom and will receive ½ an absence. Should this happen during an exam, the student will not be allowed to retake the exam at another time, while at the same time receiving a full absence.

VIII. Course Format

A. Instruction

This course utilizes both lecture/discussion and laboratory exercises. Students are expected to attend three hours of lecture/discussion per week and two hours of laboratory per week. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Once a week students should have at least 30 minutes of collaborative problem solving activity.

B. Web supplement

Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:
Course syllabus
Course assignments
Sample solutions to examinations (after being graded and returned)
Sample solutions to programming assignments (after being graded and returned)
Course calendar (an active utility)
Course e-mail (an active utility)
Course discussion list (an active utility)
Student course performance (an active utility)

Classroom Hours (5 hours per week) Class: 3
Lab: 2

IX. Topics to be Covered

A. UNIX/LINUX environment
1. Editor
2. Utilities
3. File systems
4. Links and shells
5. Shell programming

B. C programming
1. Basics
2. Control flow
3. Pointers, arrays, functions
4. File manipulation
5. Process Control

C. Python programming
1. Basics
2. Control flow
3. File manipulation
4. Task Automation
5. Standard Python Modules

X. Laboratory Exercises

This course requires a weekly 2-hour laboratory component. Topics to be covered in the laboratory sessions should include:

Lab 1- Linux Introduction and Ubuntu Installation
Lab 2- Linux commands for Files and directories manipulation
Lab 3 – Linux commands for System Users Management
Lab 4,5 – UNIX Shell Programming practice
Lab 6– Introduction to C Programming
Lab 7,8 – Pointers, memory allocation, and functions in C
Lab 9 – Introduction to Systems programming in C (file manipulation, etc)
Lab 10 - Process Control in C
Lab 11 – Introduction to Python Programming
Lab 12,13 – File manipulation and task Automation using Python
Lab 14 – Modules in Python

Three additional lab sessions should be kept in reserve to allow the instructor to extend the more difficult laboratories for more than one session.

XI. Technology Component

This course makes use of the university's wireless access infrastructure. The course relies on the university and the students having access to professional grade application development environments for the students to use. The course has a laboratory component that would be best implemented in university provided laboratory space.

XII. Special Projects/Activities

Students are required to keep a "reflective notebook" in which, after each class, they enter their own assessments of what they learned, and what questions remain from the class. From each exercise set, each student selects one problem, which the student thinks best reflects the way the topic is used in a technical context. A detailed solution to the problem is included in the student's reflective notebook.

XIII. Textbooks and Teaching Aids

A. Required Textbook

1. Sobell, M.G. and Helmke, M., Practical Guide to Linux commands, Editors and Shell Programming,4th Edition, Prentice Hall, 2017.
ISBN 978-0134774602

2. Schwartz, R. L., and B.D. Foy, Learning Perl, 7th Edition. O'Reilly, 2016.
ISBN 978-1491954324

2. Learning Python, 5th Edition Fifth Edition Edition
Publisher: O'Reilly Media; Fifth Edition edition (July 6, 2013)
ISBN-13: 978-1449355739

B. Alternative Textbooks
None.

C. Supplemental Print Materials
None.

D. Supplemental Online Materials
As available from publishers.

COSC 4361: Operating Systems

Semester Credit Hours: 3 (3,0)

I. Course Overview

This course is the study of the principles, purposes, and organization of operating systems. The goal is to prepare students an understanding of the theory as well as practices of the design and implementation of operating systems software.

II. PMU Competencies and Learning Outcomes

Students in this course develop conceptual and programming skills necessary for continued success in computer science. The skills enhance their abilities to appreciate the theory and practices of operating systems common to computer science as a discipline and to effectively communicate their solutions to fellow professionals. This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course includes individual as well as group projects, establishes both conceptual reasoning skills and technical communication skills, and provides opportunities for the presentation and defense of designed solutions.

III. Detailed Course Description

COSC 4361: The Operating Systems course is concerned with the study of the principles, purposes, and organization of operating systems, including processes, tasks, scheduling, inter-process communication, synchronization, mutual exclusion, memory management, device management, file systems, security and protection, multi-CPU systems, computer networking, and distributed computing.
This course satisfies three hours of the requirements for the degree in computer science. It is required of all students pursuing a degree program in computer science within the College of Computer Engineering and Science. It should be taken in the first semester of the senior year.

IV. Requirements Fulfilled

This course is required for all students majoring in Computer Science and Computer Engineering in the College of Computer Engineering and Science. It is also recommended as an elective for students majoring in IT.

V. Required Prerequisites

GEIT 1412: Computer Science II
GEIT 2421: Data Structure
GEIT 3331: Computer Organization
COSC 3411: Systems Programming

VI. Learning Outcomes

In this course, students will be able to:

1. Describe the underlying structure and services provided by operating systems
2. Describe and differentiate between the concepts of threads and processes
3. Explain and evaluate merits of different process scheduling algorithms
4. Explain inter process communication and process synchronization issues in cooperating

processes or threads.
5. Explain the structure and design of memory and storage management in computer systems.
6. Develop the communication, leadership and teamwork skills necessary to work in, or in charge of teams.

VII. Assessment Strategy

This course is designed with three primary goals in mind: to introduce students to the conceptual basis and practical issues associated with the compiler construction, to enhance the student's programming techniques to its application in computer science, and to provide students with the opportunity to communicate their designs and implementations to their peers in a professional setting. With this in mind, the course grade involves an assessment of their performance on examinations that focus on the application of programming paradigms to the solutions of problems, the performance analysis of the designed solutions, and the communication of designed solutions to those problems to an audience. Course grades are based on:
Homework to motivate students to do the work and earn credit accordingly.
Announced and unannounced quizzes.
Midterm exam.
A comprehensive final exam.

The final grade is based on 20% credit for the quizzes, 25% on programming assignments, 20% for in-class midterm exam and 30% for the final examination. 5% marks will be awarded based on class participation.
Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he/she proceeds through the curriculum.

Evaluation

Class Participation	5 %
Programming Assignments	25 %
Quizzes	20%
Midterm Exam	20 %
Final Exam	30 %

Total 100%

Grading Scale

A+	96 - 100%
A	90 - 95 %
B+	86 – 89 %
B	80 – 85 %
C+	76 – 79 %
C	70 – 75 %
D+	66 – 69 %
D	60 – 65 %
F	Below 60 %

VIII. Course Format

This course utilizes a mix of in-class lectures, discussions, and software demonstrations designed to help students learn the various tasks involved in designing and implementation of an operating system. While class meetings are utilized to emphasize conceptual foundation in topics related to these tasks, software demonstrations are used to provide students with hands-on training in performing these tasks. Students are expected to attend three hours of lecture per week. There are no scheduled lab hours for this course.

In addition, the instructor should consider creating a Web site for this course using Web technologies

such as WebCT or BLACKBOARD. At minimum, the site should include:
Course syllabus
Lecture material (for example PowerPoint slides, lecture notes, etc.). These should be placed on the site ahead of class meeting so that students may use the material to prepare for the lecture
Software demonstration exercises completed in class
Out-of-Class assignments and projects
Mechanism for students to digitally submit their assignments
Course calendar
Mechanism to communicate electronically (for example e-mail)
Discussion groups
Students course performance measures
Classroom Hours (3 hours per week) Class: 3
Lab: 0

IX. Topics to be covered

Week	Chapters	Topics
1	Introduction/ Chapter 1	Introduction
2	Chapter1/Chapter 3	Introduction (continued), Virtual Machines (Vmware)
3	Chapter 3	Process Concept
4	Chapter 4	Multithread Programming
5	Chapter 5	Process Scheduling
6-7	Chapter 6	Synchronization
8	Chapter 7	Deadlock
9-10	Chapter 8	Memory Management/ MidTerm Exam
11-12	Chapter 9	Virtual Memory
13-14	Chapter 10, Chapter 11	File Systems
15	Chapter 2 & 12	Systems Structures /Secondary Storage Structure
16	Final Exam	(Final exam. Scheduled by the Registrar)

X. Laboratory Exercises

There is no lab component for this course. However, the students are assigned one out-of-class application development exercise every three weeks. These hands-on exercises are designed to illustrate various development concepts covered during lecture meetings. Thus, students are expected to complete about four such exercises. Collectively, these exercises enable students to learn the details of the operating systems. The following major areas should be covered in these exercises:
Process and thread management
Process synchronization
Memory management
File system managemen

XI. Technology Component

A. In class, the instructor makes use of state-of-the art multimedia projection equipment and software. These are used to project slides and Web-based content relevant to the concepts of operating systems.
B. Outside class, the instructor uses Web-based course management software (for example WebCT, BLACKBOARD) to interact with students as described under course format section.

XII. Special Projects/Activities

There will be multiple programming based mini-projects in this course. These projects are designed to permit students to apply concepts, methods, and tools learned in class. The students are expected to:
Use operating system calls to create and manage processes
Use operating system calls to create and manage thread
Use system calls for memory and file system management

XIII. Textbooks and Teaching Aids

A. Required Textbook

Silberschatz, A., Operating Systems Concepts. 9th Edition, Wiley, 2012.
ISBN: 978-8126554270

B. Alternative Textbooks

Tanenbaum, A., Modern Operating Systems, 4th Edition, Prentice Hall, 2014.
ISBN 978-1292061429.

C. Supplemental Print Materials

As available from publisher.

D. Supplemental Online Materials

As available from publisher.

COSC 4461: Programming Languages

Prerequisites

- GEIT 1412: Computer Science II
- COSC 3411: Systems Programming

Credit hours

4 Credit Hours

Office and Phone

Room:
Tel: 3-849-
E-mail: @pmu.edu.sa

Office Hours: TBA

Course overview

Programming languages is the study of basic concepts and constructs underlying the design of the modern programming languages. Various programming paradigms, including object-oriented, functional, logic, and concurrent programming, are discussed.

PMU Competencies and Learning outcomes

Students in this course develop skills necessary for understanding the design of the modern programming languages so as to appreciate the strengths and limitations among different programming paradigms. These skills are necessary for continued success in computer science. This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course includes a structured laboratory component to ensure that students gain the necessary experience and skill in managing the concepts introduced in the class. The course includes individual as well as group projects and provides opportunities for the presentation and defense of designed solutions.

Detailed Course Description

COSC 4461: Programming Languages is concerned with the study of basic concepts, including types, control structures, abstraction mechanisms, inheritance, concurrency, and constructs underlying the design of the modern programming languages. Various programming paradigms, including object-oriented, functional, logic, and concurrent programming, are discussed. Students are exposed to the implementation of programs using programming languages under different programming paradigms. One important lasting effect of this course is to enhance and develop the ability to design, implement and test solutions to effectively solve programming problems utilizing appropriate programming languages.

Learning Outcomes

In this course, students learn:

- To develop an understanding of various basic concepts and constructs underlying the design of modern programming languages.
- To earn experience in designing and testing solutions to programming problems implemented with various programming paradigms.
- To be able to discuss the strengths and limitations of different programming paradigms in solving programming problems.
- To develop improved communication and collaborative skills.

Required Text

A. Required Textbook

Scott, M.L., Programming Language Pragmatics, 4th edition, Morgan Kaufmann, 2015.

ISBN 978-0124104099

Sethi, R.. Programming Languages: Concepts and Constructs. _____: Addison-Wesley,1995.
ISBN 0-201-59065-4

B. Alternative Textbooks

Krishnamurthi, S. Programming Languages: Application and Interpretation. _____: _____, 2003.
Available at: http://www.cs.brown.edu/~sk/Publications/Books/ProgLangs

C. Supplemental Print Materials

None.

D. Supplemental Online Materials

As available from the publisher.

Class Rules

- Disruptive Behavior
  - Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
  - Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

- Class attendance

Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness

When a student is late for 3 times it is counted as one absent.

Withdrawal

It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment

Students are assessed based on: their performance in two exams (midterm and final); their class participation, which includes the discussion of recent articles taken from online industry publications; and the quality of a final team project and related oral presentation. The relative weights of each of these items on the final grade are as follows:

The midterm and final exams account for 45% of the grade.

- Class participation and homeworks account for 15% of the grade, and is evaluated based on the ability of students to add to the material already provided by the instructor to them.
- The final team project accounts for 15% of the grade. It is evaluated based on a project document, oral presentation, and client perceptions of the team project. The project must be conducted in collaboration with a client organization (for example, a department at a large company or non-profit organization). A letter from the main contact person at the client organization, discussing and evaluating the project and its outcomes, must be provided to the instructor. The letter should contain the contact information of the person writing so the instructor can call him/her up and inquire about the project.
- Lab exercises account for 25%. They are evaluated based on their practical expertise during lab assignments.

The exams encourage the students to review all of the concepts and methods discussed in class, which are primarily based on textbook material. This is complemented by the class discussions on recent articles taken from online industry publications, which allow the students to become conversant with the industry-specific lingo related to database design issues. The final project provides an experience where concepts, methods, and industry-relevant issues are all brought together in a very applied manner to solve a real problem faced by a real organization. While this project is not as extensive as a program capstone project, it gives the students the necessary exposure to industry-relevant issues to prepare them for the future challenge of conducting a final program capstone project, and subsequently pursuing a successful career as IT professionals.

Lab Exercises	**25%**
Homework Assignments	**10%**
Midterm Exam	**15%**
Project	**15%**
Final Exam	**30%**
Class Participation	**5%**
Total	**100%**

Course Format

A. Instruction

This course utilizes both lecture/discussion and laboratory exercises. Students are expected to attend three hours of lecture/discussion per week. Students should be prepared each week to collaborate at the design and devlopment of applications as long as they rely on concepts already presented in class.

B. Web supplement

Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:

- Course syllabus
- Course assignments
- Sample solutions to examinations (after being graded and returned)
- Sample solutions to programming exercises (after being graded and returned)
- Course calendar (an active utility)
- Course e-mail (an active utility)
- Course discussion list (an active utility)
- Student course performance (an active utility)

Laboratory Exercises

This course requires a weekly 2-hour laboratory component. Topics to be covered in the laboratory sessions should include:

- Compilers – Exercises in using various programming language compilers in UNIX.

- make Utility – Exercises in using the UNIX make utility.
- SCHEME I – Exercises in familiarizing the use of the functional programming language SCHEME.
- SCHEME II – Additional exercises in familiarizing the use of SCHEME.
- Prolog I – Exercises in familiarizing the use of the logic programming language Prolog.
- Prolog II – Additional exercises in familiarizing the use of Prolog.
- Java I – Exercises in familiarizing the use of Java.
- Java II – Additional exercises in familiarizing the use of Java.
- PYTHON - Exercises in familiarizing the use of the interpreted OO programming language PYTHON.
- List Manipulation – Exercises in performing list manipulations in SCHEME.
- Unification – Exercises in performing unifications in Prolog.
- Concurrency – Exercises in exploiting concurrency in Java.
- Three additional lab sessions should be kept in reserve to allow the instructor to extend the more difficult laboratories for more than one session.

CLASS SCHEDULE (Tentative)

IX. Topics to be Covered

A. Variety of programming languages
   A. Abstraction
   B. Compilers and interpreters
   C. Syntax and semantics
   D. Context-free grammars
B. Imperative languages
   A. Control structures
   B. Data types and their presentation
   C. Composite types
   D. Subprograms, functions, procedures, methods
   E. Program structure
   F. Exception handling
C. Programming paradigms
   A. Object-oriented programming
   B. Functional programming
   C. Logic programming
   D. Concurrent programming

Week	Lecture	Chapters	Topics
1		Chapter 1	The Role of Programming Languages
2		Chapter 2	Language Description: Syntactic Structure
3		Chapter 2	Context-Free Grammars, Grammars for Expressions
4		Chapter 3	Structured Programming, Syntax, Control Flow
5		Chapter 4	Data Representation, Types, Arrays, Pointers
6		Chapter 5	Procedure Activation, Parameter-Passing Methods
7		Chapter 6	Object Oriented, Grouping of Data and Operations
8		Chapter 7	Objects, Inheritance
Semester Break			
9		Chapter 8	Functional Programming, Types, Expression Evaluation
10-11		Chapter 9	Functional Programming: Lists
12-13		Chapter 11	Logic Programming, Prolog
14-15		Chapter 12	Prolog, Data Structures, Programming Techniques

COSC 4362: Artificial Intelligence

Course Information

Course number		**COSC 4362**
Course name		**Artificial Intelligence**
Credit hours		**3**
Course Prerequisites		COSC 3421 – Data Structures COSC 3351 – Algorithms

Course Overview

The course presents an overview of artificial intelligence and its methods for solving problems. Basic algorithms for finding solutions to problems or adaptively improving responses to situations are discussed. Expert systems, genetic algorithms, and intelligent agents are among the areas that are explored.

PMU Competencies and Learning Outcomes

Students of COSC 4362: Artificial Intelligence develops skills necessary for understanding the design of artificial intelligence applications so as to appreciate the strengths and limitations of artificial intelligence approaches to problem solving. This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course is primarily a lecture-based course with the student required to complete significant projects outside of class time. The course includes individual as well as group projects (e.g. raspberryPi's AI project) and provides opportunities for the presentation and defense of designed solutions. The course encourages the development of professional communication skills and provides opportunities for collaborative project development.

Detailed Course Description

This course examines artificial intelligence as a tool for solving problems. The course looks at the major categories (application domains) of artificial intelligence (AI), including vision, natural language, planning and others with a view to identifying characteristics that mark domains as potential areas of interest. An overview of search algorithms is presented with an emphasis on heuristic driven search, including the A* algorithm and game trees. The course discusses approaches to knowledge representation, including those based on predicate calculus and non-symbolic representations like neural networks. It introduces the concepts of expert systems, intelligent agents and genetic algorithms.

Learning Outcomes

In this course, students learn:

CLO1 Outline and explain various basic concepts and constructs underlying artificial intelligence

CLO2 Design and test AI solutions to programming problems.

CLO3 State and illustrate various AI developmental tools

CLO4 Assess critically the various presented AI techniques and apply them to real world problems

CLO5 Develop improved communication and collaborative skills.

Topics to be covered (tentative, subject to change)

Week 1-2		**Introduction to AI** Chapters 1-2. AI: Definition, history and applications
Week 3-4-5		**Knowledge representation and automated Reasoning** Representing knowledge, propositional logic, first-order predicate logic, automated deduction, resolution
Week 6-7		**Strong method problem solving** Rule-based expert systems, model-based and case-based reasoning
Week 8-9		**Uncertain Reasoning** Fuzzy logic: representation and inference.
Week 10		**Problem Solving** Uninformed search, informed (heuristic) search, game playing.
Week 11-12		**Machine Learning** Inductive learning for classification, decision-tree induction, neural-networks: representation and training.
Week 13-14		**Natural Language Processing** Syntactic, semantic, and pragmatics analysis. Resolving ambiguity.
Week 15		Final Exams

Educational Resources
Required Textbook

Required Textbook

A. Russell, S. and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd Edition, Pearson, 2009.

ISBN 978-9332543515

B. Luger, G.F. Artificial Intelligence: Structures and Strategies for Complex Problem Solving. _____: Pearson Addison Wesley, 2002.

ISBN: 0-201-64866-0.

B. Alternative Textbooks

Norvig, P. Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp. _____: Morgan Kaufmann, 1991.

ISBN: 1-558-60191-0

Assessment Strategy

Class participation	10 %
Assignments & presentations	15 %
Group Projects	20%
Major 1	15 %
Major 2	15 %
Final Exam	25 %
Total	100%

1. Grading Scale

A+	**96 - 100%**
A	**90 - 95 %**
B+	**86 – 89 %**
B	**80 – 85 %**
C+	**76 – 79 %**
C	**70 – 75 %**
D+	**66 – 69 %**
D	**60 – 65 %**
F	**Below 60 %**

Attendance & Make-up exams

1. Absences during the DROP/ADD week count.

2. Late submissions – score reduced by 10 % each day after due date. No work will be accepted after one week of the due date.

3. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

4. Tardiness: when a student is late for 3 times, it is counted as one absent.

5. There is no make-up exam except for genuine medical emergencies.

6. It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Academic Honesty and Integrity

1. The instructor strictly adheres to all university policies regarding academic integrity.

2. Academic dishonesty includes but not limited to the following:

   i. Cheating on examination or other academic work,

   ii. Plagiarism and

   iii. Collusion which means unauthorized collaboration with another in preparing work offered for academic credit.

3. Academic dishonesty will not be tolerated and the PMU academic regulations will be strictly applied.

Unless specifically expressed by the instructor, collaboration between students in this course, between students in previous courses, external assistance in any form or presenting resources/research without proper citation which has been developed by another individual or organization is strictly prohibited. ALL WORK MUST BE THE RESULT OF YOUR OWN EFFORTS.

COSC 4363: Theory of Computation

Semester Credit Hours: 3 (3,0)

I. Course Overview

This course is to give an introductory study of automata, formal languages, and computability, including set theory and countability, finite automata and regular languages, push-down automata and context-free languages, Turing machines, Church's thesis, halting problem, and uncomputability.

II. PMU Competencies and Learning Outcomes

Students in this course develop quantitative skills necessary for continued success in computer science. The skills enhance their abilities to analyze and comprehend mathematically the design specifications of programming problems common to computer science as a discipline and to effectively communicate their solutions to fellow professionals. This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course includes individual as well as group projects, establishes both mathematical reasoning skills and technical communication skills, and provides opportunities for the presentation and defense of designed solutions.

III. Detailed Course Description

COSC 4363 is concerned with the an introductory study of automata, formal languages, and computability, including set theory and countability, finite automata and regular languages, push-down automata and context-free languages, Turing machines, Church's thesis, halting problem, and uncomputability. One important lasting effect of this course is to develop the ability and reasoning to understand mathematically the design specifications of programming problems presented in this course.

IV. Requirements Fulfilled

COSC 4363: Automata Theory satisfies three hours of the requirements for the degree in computer science. It is required of all students pursuing a degree program in computer science within the College of Computer Engineering and Science. It should be taken in the second semester of the junior year.

V. Required Prerequisites

- COSC 3351: Algorithms
- MATH 2331: Linear Algebra
- MATH 2332: Differential Equations

VI. Learning Outcomes

In this course, students learn:

- To develop the basic knowledge of finite automata and regular languages.
- To be able to apply push-down automata and context-free languages.
- To be able to design Turing Machine model and apply it to specific problems.
- To understand and appreciate the issues of computability.
- To develop improved communication and collaborative skills.

VII. Assessment Strategy

This course is designed with three primary goals in mind: to further students the understanding of automata theory, to lead students to connect the mathematics to its application in computer science, and to provide students with the opportunity to communicate their ideas and their expertise to the professional community. With this in mind, the course grade involves an assessment on in-class quizzes and examinations that focus on the applications of automata theory to computer science. Course grades are based on:

- Weekly assigned homework to motivate students to do the work and earn credit accordingly.
- Weekly in-class quizzes.
- Weekly, in-class presentations by students of solutions to real world problems related to the course material and classroom discussion and critique of the presentation.
- Two in-class examinations to assess the student's accumulative mastery of content covered prior to the time of the examination.
- A comprehensive final examination to assess the student's accumulative mastery of course material.

Homework		10%
Class participation		5%
Midterm	25%	
Project	10%	
Quizzes	20%	
Final	30%	
Total	100%	

Students are required to maintain a journal of thoughts and commentaries during the course. The journal contains daily entries including the identification of areas of interest and concern, notes on the preparation of presentation and comments and analysis of classmate's presentations. The journal is reviewed weekly by the instructor to provide feedback to the students.

Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he proceeds through the curriculum.

VIII. Course Format

A. Instruction

This course utilizes lecture as the primary instruction. Students are expected to attend three hours of lecture per week. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Once a week students should have at least 30 minutes of collaborative problem solving activity.

B. Web supplement

Course home page (the university's Web tool, WebCT or BLACKBOARD) should contain the following:

- Course syllabus.
- Course assignments.
- Sample solutions to examinations (after being graded and returned).
- Sample solutions to programming assignments (after being graded and returned).
- Course calendar (an active utility).
- Course e-mail (an active utility).
- Course discussion list (an active utility).
- Student course performance (an active utility).

Classroom Hours (3 hours per week) Class: 3
Lab: 0

IX. Topics to be Covered

A. Set, relations, and functions
B. Regular languages and finite automata
   a. Deterministic finite automata (DFAs)
   b. Non-deterministic finite automata (NFAs)
   c. Regular languages
   d. Closure properties
   e. Pumping lemmas
C. Context-free languages (CFLs) and pushdown automata (PDAs)
   a. CFLs, PDAs, and their equivalence
   b. Regular grammars
   c. Closure properties
   d. Pumping lemmas
D. Turing machines
   a. Definitions and examples
   b. Turing decidability, computability, and acceptability
   c. Combining Turing machines
   d. Turing machine variants
   e. Universal Turing machines
   f. Church's Thesis
E. Undecidability

X. Laboratory Exercises

This course does not offer a separate laboratory to students.

XI. Technology Component

This course makes use of the university's wireless access infrastructure. The course relies on the university and the students having access to professional grade application development environments for the students to use.

XII. Special Projects/Activities

Students are required to keep a "reflective notebook" in which, after each class, they enter their own assessments of what they learned, and what questions remain from the class. From each exercise set, each student selects one problem, which the student thinks best reflects the way the topic is used in a technical context. A detailed solution to the problem is included in the student's reflective notebook.

XIII. Textbooks and Teaching Aids

A. Required Textbook

H. Lewis and C. Papadimitriou, Elements of the Theory of Computation, (1997) Prentice Hall
ISBN 0-12-03293-7

B. Alternative Textbooks

J. E. Hopcroft, R. Motwani, and J. D. Ullman, Introduction to Automata Theory, Languages, and Computation (2001), Addison-Wesley
ISBN 0-201-44124-1

C. Supplemental Print Materials

None

D. Supplemental Online Materials

As available from publishers.

COSC 4364: Compilers

Semester Credit Hours: 3 (3,0,3)

I. Course Overview

This course is the study of the theory and practice of constructing a compiler, including lexical analysis, parsing, semantic analysis, run-time organization, code generation, and optimization. During the course of the semester, the students complete a significant compiler project.

II. PMU Competencies and Learning Outcomes

Students in this course develop conceptual and programming skills necessary for continued success in computer science. The skills enhance their abilities to appreciate the theory and practices of compiler construction common to computer science as a discipline and to effectively communicate their solutions to fellow professionals. This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course includes individual as well as group projects, establishes both conceptual reasoning skills and technical communication skills, and provides opportunities for the presentation and defense of designed solutions.

III. Detailed Course Description

COSC 4364 is concerned with the study of the theory and practice of constructing a compiler, including lexical analysis, parsing, semantic analysis, run-time organization, code generation, and optimization. During the course of the semester, the students complete a significant compiler project. The techniques studied in this course are particularly useful for programming applications that are the input of other programs. Thus, the course enriches students to formulate, design, and implement solutions to open programming problems.

IV. Requirements Fulfilled

COSC 4364: Compilers satisfies three hours of the ELECTIVE requirements for the degree in Computer Science.

V. Required Prerequisites

- COSC 3351: Algorithms

- COSC 4461: Programming Languages

VI. Learning Outcomes

- To understand and apply the phases in compilation process.
- To understand the concept of lexical analysis and apply the tools for lexical analysis.
- To acquire the concept of parsing and apply the tools for parsing.
- To learn and apply the concept of semantic analysis.
- To understand the run-time environments in compilation.
- To acquire the knowledge of code generation and optimization.
- To develop improved communication and collaborative skills.

VII. Assessment Strategy

This course is designed with three primary goals in mind: to introduce students to the conceptual basis and practical issues associated with the compiler construction, to enhance students' programming techniques to its application in computer science, and to provide students with the opportunity to communicate their designs and implementations to their peers in a professional setting. With this in mind, the course grade involves an assessment of their performance on examinations that focus on the application of programming paradigms to the solutions of problems, the performance analysis of the designed solutions, and the communication of designed solutions to those problems to an audience. Course grades are based on:

- Weekly assigned homework to motivate students to do the work and earn credit accordingly.
- Weekly, in-class presentations by students of solutions to real world problems related to the course material and classroom discussion and critique of the presentation.
- One in-class examination to assess the student's accumulative mastery of content covered prior to the time of the examination.
- 5 programming assignments testing students understanding of the major concepts introduced during the course.
- A comprehensive final examination to assess the student's accumulative mastery of course material.

The final grade is based on 10% credit for the homework, 10% for the presentations and participation in classroom discussion, 10% on in-class examinations, 50% on programming assignments, and 20% for the final examination.

Students are required to maintain a journal of thoughts and commentaries during the course. The journal contains daily entries including the identification of areas of interest and concern, notes on the preparation of presentation and comments and analysis of classmate's presentations. The journal is reviewed weekly by the instructor to provide feedback to the students.

Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he or she proceeds through the curriculum.

VIII. Course Format

This course utilizes both lecture/discussion and laboratory exercises. Students are expected to attend three hours of lecture per week. At least once per week students should be prepared to make

presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Once a week, students should have at least 30 minutes of collaborative problem-solving activity.

Classroom Hours (3 hours per week) Class: 3
Lab: 0

Web supplement:

- Course home page on the University's BLACKBOARD system should contain the following:
- Course syllabus
- Course assignments
- Sample solutions to examinations (after being graded and returned)
- Sample solutions to programming assignments (after being graded and returned)
- Course calendar (an active utility)
- Course e-mail (an active utility)
- Course discussion list (an active utility)
- Student course performance (an active utility)

IX. Topics to be Covered

A. Phases in compilation process
B. Lexical analysis
   1. Regular expressions and finite state automata (FSAs)
   2. Non-deterministic finite state automata (NFAs)
   3. Subset construction
   4. Thompson's construction
C. Parsing
   1. Bottom-up parsing
   2. LR parsing algorithm
   3. Semantic actions
   4. Abstract syntax trees
   5. SLR parser
   6. Top-down parsing
D. Semantic analysis
   1. Scopes
   2. Symbol tables
   3. Type-checking
E. Run-time organization
   1. Scoping and storage allocation
   2. Activation records
   3. Non-Local references
   4. Parameter-passing techniques
F. Code generation
   1. Intermediate code generation
   2. Final code generation
G. Machine-independent optimizations

IX. Laboratory Exercises

This course does not offer a separate laboratory to students.

X. Technology Component

This course makes use of the university's wireless access infrastructure. The course relies on the university and the students having access to professional grade application development environments for the students to use.

XII. Special Projects/Activities

Students are required to keep a "reflective notebook" in which, after each class, they enter their own assessments of what they learned, and what questions remain from the class. From each exercise set, each student selected one problem, which the student thinks best reflects the way the topic is used in a technical context. A detailed solution to the problem is included in the student's reflective notebook.

XIII. Textbooks and Teaching Aids

A. Required Textbook

Parsons, T. W., Introduction to Compiler Construction, W. H. Freeman, 1992.
ISBN 0-7167-8261-8

B. Alternative Textbooks

Appel, A., Modern Compiler Implementation in C (or Java) (or ML), Cambridge University Press, 1998.
ISBN 0-521-60765-5 (C) or 0-521-58388-8 (Java) or 0-521-60764-7 (ML)

B. Supplemental Print Materials

None.

C. Supplemental Online Materials

As available from publisher.

COSC4373: Computer Vision

Semester Credit Hours: 3 (3,0)

I. Course Overview

How can computers understand the visual world of humans? How should we develop and apply the existing algorithms so that computer can 'understand' or 'recognize' objects in a given image. This course deals with these kinds of issues where a perception is formed from a given image.

II. PMU Competencies and Learning Outcomes

Introduction to the basic concepts in computer vision. Students should be able to perform basic image analysis using algorithmic techniques. Students learn how to use MATLAB or OpenCV to implement algorithms related to computer vision.

III. Detailed Course Description

First, an introduction to low-level image analysis methods, including image formation, edge detection, feature detection, and image segmentation. Image transformations (e.g., warping, morphing, and mosaics) for image synthesis. Methods for reconstructing three-dimensional scene information using techniques such as depth from stereo, structure from motion, and shape from

shading. Motion and video analysis. Three-dimensional object recognition.

IV. Requirements Fulfilled

This course is required for computer science students. It is also an elective for IT students. It should be taken in the final year.

V. Required Prerequisites

MATH 2313: Probability & Statistical Methods

GEIT 2421: Data Structures

VI. Learning Outcomes

In this course, students learn:

1) To develop the concepts related to Images, cameras, and image formation.
2) To implement techniques for detecting edges, filtering, and texture.
3) To apply techniques for the analysis of Optical flow (image motion): affine flow, regression, dense flow.
4) To learn the techniques related to Stereo, Matching, and Tracking.
5) Develop the communication, leadership and teamwork skills necessary to work in, or in charge of teams

VII. Assessment Strategy

Assignments	30 %
Midterm Exam	25 %
Quizzes	10 %
Class participation	5 %
Final Exam	30 %
Total	100 %

Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he or she proceeds through the curriculum.

VIII. Course Format

This course utilizes both lecture/discussion and homework exercises. Students are expected to attend three hours of lecture/discussion per week. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Once a week students should have at least 30 minutes of collaborative problem solving activity.

Classroom Hours (3 hours per week) Class: 3

Web supplement: Course home page (the university's Web tool, WebCT or BLACKBOARD) should contain the following:

- Course syllabus
- Course assignments
- Sample solutions to examinations (after being graded and returned)
- Sample solutions to programming assignments (after being graded and returned)
- Course calendar (an active utility)
- Course e-mail (an active utility)

- Course discussion list (an active utility)
- Student course performance (an active utility)

IX. Topics to be Covered

- Images, cameras, and image formation
- Camera Modeling and Calibration
- Image statistics, edges, filtering, and texture
- Segmentation and grouping
- Color and Texture
- Fitting: Hough Transform
- Optical flow (image motion): affine flow, regression, dense flow
- Stereo
- Matching*
- Tracking*
- Principal component analysis and Eigen-models of objects

X. Laboratory Exercises

N/A

XI. Technology Component

This course makes use of the university's wireless access infrastructure. The course relies on the university and the students having access to professional grade application development environments for the students to use.

XII. Special Projects/Activities

Students are required to keep a "reflective notebook" in which, after each class, they enter their own assessments of what they learned, and what questions remain from the class. From each exercise set, each student selects one problem, which the student thinks best reflects the way the topic is used in a technical context. A detailed solution to the problem is included in the student's reflective notebook.

XIII. Textbooks and Teaching Aids

A. Required Textbook
Computer Vision-A Modern Approach, Forsyth & Ponce, Prentice Hall 2003.

B. Alternative Textbooks
Introductory Techniques for 3-D Computer Vision, Trucco & Alessandro Verri, Prentice Hall, 1998.

D. Supplemental Online Materials

As available from publisher.

OFFICE AND PHONE

Dr. Shahabuddin Muhammad
Room: S-008
Tel: 03-849-9752
E-mail: smuhammad@pmu.edu.sa

COSC 3354: Introduction to Cryptography

Semester Credit Hours: 3 (3, 0)

I. Course Overview

This course introduces the main types of cryptographic mechanisms, and explains how different cryptographic mechanisms provide various security services. This course will also identify some key issues relating to the management of these services.

II. PMU Competencies and Learning Outcomes

This course has both theoretical and practical aspects. Students develop principles behind modern cryptography. They gain an understanding of and appreciation for cryptographic applications in network security. This course makes an extensive use of the PMU infrastructure to provide communication between faculty and students. While the course does not include a structured laboratory component, practical assignments/projects are assigned to ensure students gain relevant practical exposure.

III. Detailed Course Description

This course is an in-depth introduction to cryptology, covering material from classic to modern encryption methods. It explores how secret messages were hidden historically along with their weaknesses. It also introduces the basic code breaking techniques. Modern symmetric and asymmetric ciphers such as 3DES, AES, RSA, and DH will be discussed. This course also covers relevant mathematical concepts like modular arithmetic and number theory.

IV. Requirements Fulfilled

This course satisfies three hours of the requirements for degrees in Computer Science and Computer Engineering. It is an available elective for the degree in Information Technology. It should be taken no earlier than the junior year.

V. Required Prerequisites

GEIT 2331 Mathematical reasoning and Algorithmic Thinking

VI. Learning Outcomes

In this course, students learn:

CLO1: Explain basic cryptography concepts, terminology and issues.
CLO2: Develop the required knowledge and skills to be able to use symmetric key encryption
CLO3: Develop the required knowledge for understanding message integrity, authentication and digital signature
CLO4: Develop the required knowledge and skills to be able to use public key cryptography (asymmetric key encryption)
CLO5: Demonstrate the communication, leadership and teamwork skills necessary for effectively working as professionals in teams, or in charge of teams

VII. Assessment Strategy

Course grades are be based on

- Regularly assigned homework to motivate students to do the work and earn credit accordingly.
- Weekly structured in-class exercises designed to guide students through specific course topics.
- Several in-class quizzes/exams to assess students' accumulative mastery of content covered prior

to time of exam.
- A comprehensive final exam to assess students' accumulative mastery of course material.

Students' final grades are based on 30% credit for homework and projects, 5% for participation in classroom discussion, 25% on major exams, 15% on quizzes and 25% for the final examination.

Students are required to maintain a journal of thoughts and commentaries during the course. The journal contains daily entries including the identification of areas of interest and concern, notes on the preparation of presentation and comments and analysis of classmate's presentations. The journal is reviewed weekly by the instructor to provide feedback to the students.

Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he proceeds through the curriculum.

VIII. Course Format

This course utilizes both lecture/discussion and in-class exercises. Students are expected to attend three hours of lecture/discussion per week. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Once a week students should have at least 30 minutes of collaborative problem solving activity.

Classroom Hours (6 hours per week) Class: 3
Lab: 0

Web supplement: Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:

- Course syllabus
- Course assignments
- Sample solutions to examinations (after being graded and returned)
- Course calendar (an active utility)
- Course e-mail (an active utility)
- Course discussion list (an active utility)
- Student course performance (an active utility)

IX. Topics to be covered (Not in the same order)

1. Core security services provided by cryptography
2. Basic model of a cipher system and use of cryptography
3. Historical algorithms
4. Symmetric-key encryption
5. Attacks on Ciphers
6. Data integrity
7. Entity authentication
8. Introduction to modulo arithmetic
9. Public key encryption
10. Digital signatures
11. Cryptographic protocols
12. Key management
13. Cryptographic applications

X. Laboratory Exercises

N/A

XI. Technology Component

This course makes use of the university's wireless access infrastructure during the class/lecture sessions. The course relies on the university and the students having access to a professional grade computing environment for practical exercises and demonstrations.

XII. Special Projects/Activities

1. Online Cisco course on "cyber security" as an application of cryptography
2. Codebreaking tasks based on simple classical ciphers
3. Cryptool tasks on
    a. Cryptanalysis
    b. Cryptography
    c. Hash functions

XIII. Textbooks and Teaching Aids

A. Required Textbook

Main Textbook: J. Katz and Y. Lindell, Introduction to Modern Cryptography (2nd edition).

B. Alternative Textbooks

W. Stallings, Cryptography and network security (7^{th} edition).

C. Supplemental Print Materials

D. Supplemental Online Materials

As available from publisher

COSC 4371: Computer Graphics

Semester Credit Hours: 3 (3)

I. Course Overview

Computer graphics generally deals with the creation, storage and manipulation of models and images by studying the use of computers to synthesize and manipulate visual information.

II. PMU Competencies and Learning Outcomes

This course provides an introduction to a wide range of topics in computer graphics, and prepares the students for continued studies in more specialized sub-fields.

Students will learn:

- Fundamentals of computer graphics algorithms
- Basics of real-time rendering and graphics hardware
- Basic OpenGL
- C++ programming experience

III. Detailed Course Description

COSC 4371: Computer Graphics is concerned with the basic algorithms and skills necessary for modeling and visualizing information. The course will introduce important concepts such as the following:

- computer graphics hardware
- Computer graphics applications and motivation
- OpenGL 2D and 3D
- Rendering
- Texture mapping
- Lighting
- Coordinate spaces and transformation
- Image processing & editing
- Ray tracing
- Animation
- Virtual reality
- Scientific visualization

IV. Requirements Fulfilled

This course satisfies three hours of elective course for students pursuing a degree program in computer science and computer engineering within the College of Computer Engineering and Sciences.

V. Required Prerequisites

- Good background in C/C++ or Java and relevant IDEs.

VI. Learning Outcomes

In this course, students learn:

- Knowledge of the principal basics of computer graphics
- Write programs in the in the context of computer graphics
- Use OpenGL for graphics programming (or a comparable API)
- Develop improved communication and collaborative skills

VII. Assessment Strategy

This course is designed with primary goals in mind: to introduce students to the conceptual basis and practical techniques of computer graphics, and to provide students with the opportunity to communicate their designs and implementations to their peers in a professional setting. With this in mind, the course grade involves an assessment of their performance that focus on the application of programming techniques to the solutions of problems and the communication of designed solutions to those problems to an audience. Course grades are based on:

- Assignments allowing the students to practice the techniques and testing students understanding of the major concepts introduced during the course.
- Major examinations to assess the student's accumulative mastery of content covered prior to the time of the examination.
- A comprehensive final examination to assess the student's accumulative mastery of course material.
- A group final project which would reflect the student's understanding of the course material.

Assessment:

Participation and assignments		15%
Quizzes	10%	
Major-1	15%	
Major-2	15%	
Group project		15%
Final exam	30%	

Grading Scale: Standard

A+	**96 - 100%**
A	**90 - 95 %**
B+	**86 – 89 %**
B	**80 – 85 %**
C+	**76 – 79 %**
C	**70 – 75 %**
D+	**66 – 69 %**
D	**60 – 65 %**
F	**Below 60 %**

Class Rules:

- Disruptive Behavior
  - Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement,

questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.

- Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.
- Eating / Drinking / Smoking: students are requested to refrain from engaging in these activities while in class.

- Class attendance
  - Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

- Tardiness
  - If a student is late for 3 times it is counted as one absence.

- Make-up
  - Midterm & Final exams – different from and harder than in-class exams. You should have a valid excuse as per PMU policies

- Late submissions – score reduced by 10% points per day after due date. No work will be accepted after one week of the due date.

- Withdrawal
  - It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

- Academic Honesty and Integrity
  - The instructor strictly adheres to all university policies regarding academic integrity.
  - Academic dishonesty includes but not limited to the following:
    - Cheating on examination or other academic work,
    - Plagiarism and
    - Collusion which means unauthorized collaboration with another in preparing work offered for academic credit.
  - Academic dishonesty will not be tolerated and the PMU academic regulations will be strictly applied.

- Unless specifically expressed by the instructor, collaboration between students in this course, between students in previous courses, external assistance in any form or presenting resources/research without proper citation which has been developed by another individual or organization is strictly prohibited. ALL WORK MUST BE THE RESULT OF YOUR OWN EFFORTS.

➢ Miscellaneous

Mobile phones: A student whose mobile rings during class will be asked to leave the classroom and will receive ½ an absence. Should this happen during an exam, the student will not be allowed to retake the exam at another time, while at the same time receiving a full absence.

VIII. Course Format

A. Instruction

This course utilizes lecture/discussion. Students are expected to attend three hours of lecture/ discussion per week.

B. Web supplement

Course home page (the university's Blackboard) should contain the following:

- Course syllabus
- Course assignments
- Course e-mail

Classroom Hours (3 hours per week) Class: 3

XI. Special Projects/Activities

Students are required to keep a "reflective notebook" in which, after each class, they enter their own assessments of what they learned, and what questions remain from the class.

XII. Textbooks and Teaching Aids

A. Required Textbook

1. Computer Graphics: Principles and Practice. 3rd Edition 2013

   Publisher: Addison-Wesley Professional

   ISBN-10: 0321399528

   ISBN-13: 978-0321399526

2. OpenGL Programming Guide. 8th Edition

   Publisher: Addison-Wesley, 2013.

   http://www.glprogramming.com/red/

1. B. Alternative Textbooks

Computer Graphics with OpenGL. 4th Ed

Publisher: Pearson. 4th edition (November 19, 2010)

ISBN-10: 0136053580

ISBN-13: 978-0136053583

2. Fundamentals of Computer Graphics. 3rd Edition

Publisher: A K Peters/CRC Press; 3 edition (July 21, 2009)

ISBN-10: 1568814690

ISBN-13: 978-1568814698

C. Supplemental Print Materials

None.

D. Supplemental Online Materials

As available from publishers.

COSC 4374: Computer Vision

Semester Credit Hours: 3 (3,0)

I. Course Overview

How can computers understand the visual world of humans? How should we develop and apply the existing algorithms so that computer can 'understand' or 'recognize' objects in a given image. This course deals with these kinds of issues where a perception is formed from a given image.

II. PMU Competencies and Learning Outcomes

Introduction to the basic concepts in computer vision. Students should be able to perform basic image analysis using algorithmic techniques. Students learn how to use MATLAB or OpenCV to implement algorithms related to computer vision.

III. Detailed Course Description

First, an introduction to low-level image analysis methods, including image formation, edge detection, feature detection, and image segmentation. Image transformations (e.g., warping, morphing, and mosaics) for image synthesis. Methods for reconstructing three-dimensional scene information using techniques such as depth from stereo, structure from motion, and shape from shading. Motion and video analysis. Three-dimensional object recognition.

IV. Requirements Fulfilled

This course is required for computer science students . It is also an elective for IT students. It should be taken in the final year.

V. Required Prerequisites

MATH 2331 - Linear Algebra

MATH 1423 - Calculus II

VI. Learning Outcomes

In this course, students learn:

6) To develop the concepts related to Images, cameras, and image formation.
7) To implement techniques for detecting edges, filtering, and texture.
8) To apply techniques for the analysis of Optical flow (image motion): affine flow, regression, dense flow.
9) To learn the techniques related to Stereo, Matching, and Tracking.
10) Develop the communication, leadership and teamwork skills necessary to work in, or in charge of teams

VII. Assessment Strategy

Assignments	30 %
Midterm Exam	25 %
Quizzes	10 %
Class participation	5 %
Final Exam	30 %
Total	100 %

Final grades and the student and instructor observations from reflective notebooks are included in

the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he or she proceeds through the curriculum.

VIII. Course Format

This course utilizes both lecture/discussion and homework exercises. Students are expected to attend three hours of lecture/discussion per week. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Once a week students should have at least 30 minutes of collaborative problem solving activity.

Classroom Hours (3 hours per week) Class: 3

Web supplement: Course home page (the university's Web tool, WebCT or BLACKBOARD) should contain the following:

- Course syllabus
- Course assignments
- Sample solutions to examinations (after being graded and returned)
- Sample solutions to programming assignments (after being graded and returned)
- Course calendar (an active utility)
- Course e-mail (an active utility)
- Course discussion list (an active utility)
- Student course performance (an active utility)

IX. Topics to be Covered

- Images, cameras, and image formation
- Camera Modeling and Calibration
- Image statistics, edges, filtering, and texture
- Segmentation and grouping
- Color and Texture
- Fitting: Hough Transform
- Optical flow (image motion): affine flow, regression, dense flow
- Stereo
- Matching*
- Tracking*
- Principal component analysis and Eigen-models of objects

X. Laboratory Exercises

N/A

XI. Technology Component

This course makes use of the university's wireless access infrastructure. The course relies on the university and the students having access to professional grade application development environments for the students to use.

XII. Special Projects/Activities

Students are required to keep a "reflective notebook" in which, after each class, they enter their own assessments of what they learned, and what questions remain from the class. From each exercise set, each student selects one problem, which the student thinks best reflects the way the topic is used in a technical context. A detailed solution to the problem is included in the student's reflective notebook.

XIII. Textbooks and Teaching Aids

A. Required Textbook
Computer Vision-A Modern Approach, Forsyth & Ponce, Prentice Hall 2003.

B. Alternative Textbooks
Introductory Techniques for 3-D Computer Vision, Trucco & Alessandro Verri, Prentice Hall, 1998.

D. Supplemental Online Materials

As available from publisher.

COSC 4311: Parallel Computing

Prerequisites

- COSC 3351: Algorithms

Credit hours

3 Credit Hours

Office and Phone

Room: S-015
Phone: (3) 849-
E-mail: @pmu.edu.sa
Office Hours: TBA

Course overview

This course provides a basic, in-depth look at techniques for the design and analysis of parallel algorithms and for programming them on commercially available parallel platforms. Principles of parallel algorithms design and different parallel programming models are both discussed. This course is for anyone wanting to gain proficiency in all aspects of parallel and distributed programming.

PMU Competencies and Learning outcomes

Students of COSC 4311: Parallel Computing develop skills necessary for understanding the design of parallel computing applications so as to appreciate the strengths and limitations of parallel computing approaches to problem solving. This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course is primarily a lecture-based course with the student required to complete significant projects outside of class time. The course includes individual as well as group projects and provides opportunities for the presentation and defense of designed solutions. The course encourages the development of professional communication skills and provides opportunities for collaborative project development.

Detailed Course Description

This course provides a basic, in-depth look at techniques for the design and analysis of parallel algorithms and for programming them on commercially available parallel platforms. Principles of parallel algorithms design and different parallel programming models are both discussed. This course is for anyone wanting to gain proficiency in all aspects of parallel and distributed programming. Students develop skills necessary for understanding the design and analysis of parallel and distributed algorithms, and to appreciate not only the advantages, but also the difficulties of adapting algorithms to a parallel or distributed paradigm.

The course is primarily a lecture-based course with the student required to complete significant projects outside of class time. The course includes individual as well as group projects and provides opportunities for the presentation and defense of designed solutions. The course encourages the development of professional communication skills and provides opportunities for collaborative project development.

Parallel computing is a critical component of the computing technology of the 21st century, and is likely to grow in importance with the proliferation of multiprocessor PC desktops and servers and scalable clusters of commodity workstations. This course examines the organizing principles behind parallel computing both from an architectural and an algorithmic perspective. The course consists of two parts, organized around a common set of issues relevant to all parallel systems: naming, synchronization, latency, and bandwidth. The first part discusses how modern parallel computer architectures deal with these issues, both at the small (shared memory multiprocessors) and large (scalable

multiprocessors) scales. The second part of the course discusses how the issues are dealt with in several common programming paradigms including message-passing, shared-memory, data-parallel, as well as higher-level approaches. The focus in this part of the course is on both algorithmic techniques and programming for performance.

Learning Outcomes

In this course, students learn:

- To describe various basic concepts associated with parallel computing environments.
- To outline the characteristics of different physical organizations of parallel platforms.
- To explain the effects that issues of synchronization, latency and bandwidth have on the efficiency and effectiveness of parallel computing applications.
- To design and analyze parallel algorithms for various parallel platforms.
- To gain experience in designing and testing parallel computing solutions to programming problems.
- To quantify the performance of a parallel algorithm using mathematical and measurement techniques.
- To develop good communication and collaborative skills.

Required Text

C. Required Textbook

Rauber, T., Parallel Programming for Multicore and Cluster Systems, 2nd edition, Springer, 2013.
ISBN 978-3642378003

D. Alternative Textbooks

Grama, A., Gupta, A., Karypis, G., and Kuman, V.
Introduction to Parallel Computing, 2ndEdition
Addison-Wesley, 2003.
ISBN-13: 978-0-201-64865-2

Wilkinson, B., and M. Allen.
Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers
Prentice-Hall, 2000.

Pacheco, Peter.
Parallel Programming with MPI
Morgan-Kaufmann, 1996.

E. Supplemental Print Materials
None.

F. Supplemental Online Materials
None.

class RULES

- Disruptive Behavior
  - Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
  - Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance

Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply

impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness

When a student is late for 3 times it is counted as one absent.

WITHDRAWAL

It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment

Grades shall be computed on the basis of the following:

Class participation	5%
Assignments	25%
Project	20%
Midterm Exam	20%
Final Exam	30%
Total	**100%**

Z

Course Format

A. Instruction

This course utilizes both lecture/discussion and exercises. Students are expected to attend three hours of lecture per week. At least once per week students should be prepared to make presentation on the design and implementation of a parallel solution to a problem selected by the instructor and to take part in a discussion based on that presentation.

B. Web supplement

Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:

- Course syllabus.
    - Lecture material (PowerPoint slides, lecture notes, etc.). These will be placed on the site ahead of class meeting so that students may use the material to prepare for the lecture.
- Course assignments.
- Project specifications.
- Sample solutions to examinations (after being graded and returned)

CLASS SCHEDULE (Tentative)

Week #	Topics covered	Sections to Read
1	Motivating Parallelism Implicit Parallelism: Trends in Microprocessor Architectures	1.1 2.1
2	Limitations of Memory System Performance Dichotomy of Parallel Computing Platforms	2.2 2.3
3	Physical Organization of Parallel Platforms • Architecture of an Ideal Parallel Computer: PRAM • Interconnection Networks • Combinational Circuits	2.4
4	Communication Costs in Parallel Machines Routing Mechanisms for Interconnection Networks Impact of Process-Processor Mapping and Mapping Techniques	2.5 2.6 2.7
5	Principles of Parallel Algorithm Design Decomposition Techniques Characteristics of Tasks and Interactions	3.1 3.2 3.3
6	Mapping Techniques for Load Balancing Methods for Containing Interaction Overheads Parallel Algorithm Models	3.4 3.5 3.6
7	One-to-All Broadcast and All-to-One Reduction All-to-All Broadcast and Reduction	4.1 4.2
8	All-Reduce and Prefix-Sum Operations Scatter and Gather All-to-All Personalized Communication	4.3 4.4 4.5
9	Circular Shift **Midterm Exam**	4.6
10	Sources of Overhead in Parallel Programs Performance Metrics for Parallel Systems The Effect of Granularity on Performance	5.1 5.2 5.3
11	Sorting Networks	9.2
12	Bubble Sort and its Variants Quicksort	9.3 9.4
13	Enumeration Sort	
14	Broadcasting with Selective Reduction	
15	Project Presentations and Discussions	

ITAP 3313: User interface Development

Prerequisites

GEIT 1411: Computer Science I
GEIT 1412: Computer Science II

Credit hours

3 Credit Hours

Course overview

This is an introductory course to the subject of interaction design. It covers issues like the design of interactive products to support the way people communicate and interact in their everyday and working lives plus the design of spaces for human communication and interaction.
The goal of the course is to expose the students to the various techniques and methods relating to the subject, such that they can participate actively in the production and design of products and spaces that are Usable/User-Friendly. i.e. products and spaces that are both intuitive, easy to learn, effective to use and provide an enjoyable experience
The slides from the 2nd edition of the text book can be found on blackboard.
The course ends with mobile interface design and implementation as an example of use of interaction design activities.

PMU Competencies and Learning outcomes

The goal of this course is to provide the student with a basic knowledge of human-computer interaction as a distinct discipline and to investigate specific issues involving human-computer interaction and user interface design.
The course provides opportunities for technical skill development as well as communication, collaboration and leadership skills through the maintenance of journals, detailing progress in group projects, and in-class presentations
This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students.
The course includes individual as well as group projects and establishes both conceptual reasoning skills and technical communication skills.

Detailed course description

Modern computer technology requires professionals of every computing specialty to understand both hardware and software. The interaction between hardware and software at a variety of levels also offers a framework for understanding the fundamentals of computing. The performance of future software systems will be dramatically affected by how well software designers understand the basic hardware techniques at work in a system. Thus, compiler writers, operating system designers, database programmers, and most other software engineers need a firm grounding in the principles presented in this course.
In addition, this course will cover the basic theory and concepts in the area of human-computer interaction.
The course is intended to introduce the student to the basic concepts of human- computer interaction. It will cover the basic theory and methods that exist in the field. Case studies are used throughout the readings to exemplify the methods presented and to lend a context to the issues discussed. The assignments are designed to give the student practice in a set of the more basic techniques used in the human-computer interaction discipline.
The course will begin by introducing the students to the software development process and discussing the behavioral techniques that apply at different stages of this process. It will then discuss the basic applications of these techniques: survey methods, task analyses, usability studies and prototyping. It will finish by discussing a set of innovative interfaces and new developments in human computer interaction. The implementation part of the course reviews the primary concepts of mobile programming. Some background is required in object oriented programming.

Learning Outcomes

After successfully completing the course, students will be able to:

1. List, define and analyze user interface principles and issues and their importance in product design
2. Describe basic techniques for user requirement elicitation, data collection, and analysis
3. Design and implement a user interface according to HCI design principles
4. Evaluate systems design and interfaces from an interaction perspective
5. Develop the communication, leadership and teamwork skills necessary to work in or in charge of teams

Required Text

Required Textbook

Interaction Design: Beyond Human-Computer Interaction; Jennifer Preece, Yvonne Rogers, and Helen Sharp, Wiley and Son, Inc. 2002, ISBN: 0-471- 49278-7.

Alternative Textbooks
Designing the User Interface
4th edition, Ben Shneiderman and Catherine Plaisant
Addison-Wesley, 2005
ISBN: 0-321-19786-0

Human-Computer Interaction
Third Edition by Alan Dix et al
Prentice Hall (2004).
Supplemental Online Materials
http://www.id-book.com/
http://hcibib.org/
http://hcidesigns.com/beck/capstone/

Class RULES

Disruptive Behavior

Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance

Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness

When a student is late for 3 times it is counted as one absent. Students are considered tardy if they arrive after the first 10 minutes (according to the instructor's watch).

WITHDRAWAL

It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W", after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment

There will be individual homework assignments as well as one group project broken up into several phases. Some of the assignments will require programming. There will also be frequent in-class discussions and activities in which students are expected to actively participate. There will be a midterm and a final exam.

The final grade will be calculated this way:
Participation, Assignments and Presentations: 15%
Quizzes: 10%
Major 1: 15%
Major 2: 15%

Project: 20%
Final: 25%

Participation

Attendance does not mean participation, and therefore will not contribute to the participation grade. Participation will be assessed by the instructor on the basis of student's exhibited interest to the course by answering questions in class, asking questions, and actively contributing to classroom discussion.

Grading Scale

A+	96 - 100%
A	90 - 95 %
B+	86 – 89 %
B	80 – 85 %
C+	76 – 79 %
C	70 – 75 %
D+	66 – 69 %
D	60 – 65 %
F	Below 60 %

Make-up

Small homework and in class work – no make-up
Majors & Final exams – different from and harder than the scheduled exams (only if valid excuse is available). Is offered only to students with a serious medical condition.

Late submissions

For each day that the submission is late, it is a 10% penalty.

Course Format

A. Instruction

The course is primarily a lecture-based course in which the students are required to complete significant projects outside of class time. The course will include individual assignments as well as group projects and provide opportunities for the presentation and defense of designed solutions. At least once per week students should be prepared to make presentation on the design and implementation of a solution to a problem selected by the instructor and to take part in a discussion based on that presentation. Once a week students should have at least 30 minutes of collaborative problem solving activity.

B. Web supplement

Web supplement: Course home page (the university's BLACKBOARD) should contain the following:
Course syllabus.
Course assignments.
Model programmed solutions to programming assignments (once students have completed them)
Course calendar (an active utility).
Course e-mail (an active utility).
Course discussion list (an active utility).
Students course marks. (an active utility).

C. Technology Component

In class, the instructor makes use of state-of-the art multimedia projection equipment and software. These are used to project slides and Web-based content relevant to the concepts of and use of OR techniques.
Outside class, the instructor uses Web-based course management software (for example WebCT, BLACKBOARD) to interact with students as described under course format section.
All skill-building exercise and case studies assigned in this class require students to use software packages available in Microsoft Excel or specialized OR software.
When working on case assignments, each team is set up an on-line discussion group and repository to facilitate collaboration among team members.

D. Special Project

Students are required to apply the interaction design activities to design and implement a mobile application. Each group of students chooses one problem. A prototype and a detailed document showing how the interaction design principles and activities were applied for the design and construction for the mobile application should be submitted.

CLASS SCHEDULE (Tentative)

Week#	Date	Topics covered	Sections to Read
1		What is interaction design	Ch. 1
2		Understanding and conceptualizing interaction	Ch. 2
3		Interfaces and interactions	Ch. 6
4		Data gathering	Ch. 7
5			
6		Break	
7		The process of interaction design	Ch. 9
8		Identifying needs and establishing requirements	Ch. 10
9		Design, prototyping and construction	Ch. 11
10		Introducing evaluation	Ch. 12
11		Usability testing and field studies	Ch. 14
12		User-centered design for mobile apps development	
13		Design patterns for mobile apps	
14		UI design for mobile apps	
15		The Apache Cordova framework	

ITAP 4371: E-commerce

Prerequisites

GEIT 1411: Computer Science I
GEIT 1412: Computer Science II
GEIT 2341: Database Design
ITAP 3471: Web Server Administration

Credit hours

3 Credit Hours

Course overview

The primary objective of this course is to expose students to the advanced use of information technology in the design and implementation of Web-based business applications to support e-commerce. The course presents concepts, methodology, and tools for designing, implementing, and management of e-commerce applications.

PMU Competencies and Learning outcomes

This course helps students develop proficiency in the design and development of e-commerce applications. Students develop both the conceptual basis and the practical skills in the design and implementation of Web-based applications to support the core and mission-critical Internet-based business processes of an organization. Additionally, this course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. While the course does not include a structured laboratory component, out-of-class projects are assigned to ensure students gain the necessary experience and skill developing e-commerce applications. The course includes a group project and provides opportunities for the presentation and defense of their designed solution.

Detailed course description

The course is designed to provide coverage of technologies available to design, develop, implement, and manage business information processing applications that support e-commerce. The course begins with an overview of E-Commerce concepts and technologies. It then presents client-side web programming languages such as HTML5 and CSS3. It then introduces students to the use of an integrated development environment based on J2EE and/or .Net framework (for example IBM WebSphere Studio or Visual Studio .Net) in the development of various components that make up an E-Commerce application. Next, students learn to develop server-side components using technologies such as ASP.net, JavaBeans, Java Server Pages. Connectivity of the server-side components to back-end databases is introduced next. Finally, the course presents the model-view-controller (MVC) design paradigm.. The mastery of these concepts and skills is demonstrated via an end-of-term team project.
This course may be taught using one of the two competing object-oriented technologies, namely industry-standard J2EE platform or Microsoft's .Net framework. The choice of technology should depend largely on student's programming background (Java versus Visual basic or C#), industry demands, and availability of faculty with requisite skills. For these reasons, the topical outline is left generic enough to be compatible with either technology. J2EE is currently the most commonly used framework for e-commerce applications. However, Microsoft's .Net framework is gaining momentum.

Requirements fulfilled

This course is required for all students majoring in Information Technology in the College of Computer Engineering and Science. It is also recommended as an elective for students majoring in computer science and management information systems. It should be taken no earlier than the first semester of the senior year.

Learning Outcomes

In this course, students learn:
CLO1 To demonstrate knowledge about the design, development and implementation of client-side components that constitute an e-commerce web application
CLO2 To employ frameworks (for example J2EE or .Net) to design, develop and implement server-side components that constitute an e-commerce web application
CLO3 To develop skills in using an integrated development environment (IDE) for example IBM WebSphere or MS Visual Studio .Net
CLO4 To analyze main issues relevant to the implementation of enterprise-level e-commerce web applications
CLO5 To show communication, leadership and teamwork skills necessary to work in or lead teams

Class RULES

Disruptive Behavior

- Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
- Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance

Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness

When a student is late for 3 times it is counted as one absent. Students are considered tardy if they arrive after the first 10 minutes (according to the instructor's watch).

WITHDRAWAL

It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W", after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment

There will be individual homework assignments as well as one group project broken up into several phases. Some of the assignments will require programming. There will also be frequent in-class discussions and activities in which students are expected to actively participate. There will be a midterm and a final exam.

The final grade will be calculated this way:

Programing assignments and presentations: 15%
Quizzes: 10%
Major 1: 15%
Major 2: 15%
Group project: 20%
Final: 25%

Course Format

This course utilizes a mix of in-class lectures, discussions, and software demonstrations designed to help students learn the various tasks involved in designing, installing, configuring, securing, and monitoring an e-commerce application. While class meetings are utilized to emphasize conceptual foundation in topics related to these tasks, software demonstrations are used to provide students with hands-on training in performing these tasks. For best results, this course is taught in a lab-environment where students can follow the instructor during software demonstrations and discussion of key development strategies. Initial class meetings introduce students to the theoretical concepts. Most of the later meetings are used for laboratory software demonstrations geared at helping students master skills for designing and developing various E-Commerce application components. A running case study is used for this purpose so that students see an end-to-end development cycle. Students are expected to attend three hours of lecture (includes discussion/software demonstrations) per week.

In addition, the instructor should consider creating a Web site for this course using Web technologies such as WebCT or BLACKBOARD. At minimum, the site should include:
Course syllabus
Lecture material (for example PowerPoint slides, lecture notes, etc.). These should be placed on the site ahead of class meeting so that students may use the material to prepare for the lecture

Out-of-Class assignments and end-of-term project
Keys to exams (after students have completed them)
Mechanism for students to digitally submit their assignments
Course calendar
Mechanism to communicate electronically (for example e-mail)
Discussion groups

Topics to be covered

A. Developing client-side components
   1. Introduction to HTML5
   2. HTML5 elements
   3. HTML5 tables and forms elements

B. Formatting web pages using CSS3

C. Development of server-side components
   1. Introduction to ASP.Net
   2. Visual Studio Express for Web
   3. ASP.Net server Controls
   4. ASP.Net validation controls
   5. ASP.Net and Database connectivity (ADO.Net)

D. Test, debug and deploy an E-commerce application
   1. Test, debug and deploy D.
   2. Unit testing
   3. Debugging tools
   4. Deployment environment to include Web and application servers

E. Securing e-commerce application
   1. Identify threats and vulnerabilities
   2. Understand various aspects of security such as authentication and authorization.
   3. Test and deploy secured application

Technology component

A. In class, the instructor makes use of state-of-the art multimedia projection equipment and software. These are used to project slides and Web-based content relevant to the design of E-Commerce applications.
B. In class, in the laboratory setting, the instructor makes use of a commercial integrated development environment based on J2EE or .Net framework to demonstrate various tasks involved in successful design and development of an enterprise-level, mission-critical E-Commerce application.
C. Outside class, the instructor uses Web-based course management software (for example WebCT, BLACKBOARD) to interact with students as described under course format section.

Special projects/Activities

An end-of-term team project is required in this course. This project is designed to permit students to apply concepts, methods, and tools learned in class to support a real system. The project requires students, working in teams, to design and implement a complete Web-based, n-tiered e-commerce application of reasonable complexity. This may be a simulated application designed to expose students to tasks involved in assembling a real-system or it may be a real application for a real client. In either case, student teams are expected to:
Design, test, and deploy the application,
Prepare a complete technical manual, and
Make an oral presentation of their design architecture to a panel of judges comprising of the instructor and other faculty members or client.

The final project is thus designed to assess competency in performing various tasks related to the design, development and implementation of a complete E-Commerce application.

Required Text

A. Required Textbook

1. If the course is based on .Net framework:
   Imar Spaanjaars; Beginning ASP.NET 4.5.1: in C# and VB; Wrox; 2014
   ISBN: 978-1-118-84677-3 (Available on PMU's bookstore)
2. If course is based on J2EE platform:
   TBA

B. Alternative Textbooks

1 If the course is based on J2EE platform:
2. If the course is based on .Net framework:
   Jason N. Gaylord, Christian Wenz, Pranav Rastogi, Todd Miranda,
   Scott Hanselman; Professional ASP.NET 4.5 in C# and VB;
   John Wiley & Sons, 2013
   ISBN: : 978-1-118-31182-0

ITAP 4316: Introduction to Software Project Management

Semester Credit Hours: 3 (3,0)

I. Course Overview

This course will introduce the concepts and techniques for managing software projects. The students will learn the basic project management concepts including process groups, knowledge areas and project management lifecycles with special focus on Agile approaches. The students will learn techniques for initiating, planning, launching and monitoring software projects. They will also learn tools and techniques related to software configuration and build management. They will learn software project management tools and use it to manage their software project.

I. PMU Competencies and Learning Outcomes

This course helps students develop the ability to become conversant with software project management topics and understand the related terms and issues that are important for software engineering practitioners around the world. Additionally, the course provides the students with the communication, leadership, and teamwork skills necessary to effectively work as professionals in teams, or in charge of teams, responsible for software engineering projects. It introduces the students to software project management tools and its applications. It provides students with an understanding of software configuration management and software build management issues.

II. Detailed Course Description

The course introduces the project management topics that are relevant in the context of software engineering projects. This covers several fundamental project management issues such as task decomposition and related budgeting, project charting, project scope management, time and cost management, human resources management, communication management, and project risk management. The course also includes a discussion of advanced issues in connection with software engineering projects, such as emerging tools for software engineering (state of the art project, configuration and build management tools are introduced). This course emphasizes both software design/implementation issues and software engineering project management issues. This dual emphasis orientation is aimed at providing students with a realistic view of issues related to real software engineering projects, which are more often than not complex collaborative projects with a clear expectation of organizational impacts in terms of quality, productivity and/or competitiveness enhancements.

III. Requirements Fulfilled

This course is elective for all students majoring in Information Technology, Computer Science and Computer Engineering departments in the College of Computer Engineering and Science.

IV. Required Prerequisites

GEIT 1411: Computer Science I
GEIT 1412: Computer Science II
GEIT 3351: Software Engineering I

VI. Learning Outcomes

In this course, students learn to:
Explain and justify the concepts and terms related to software project management
Select, apply and analyse the appropriate software development life cycle for managing a project
Design a project plan and schedule
Develop an effective reporting system for monitoring and controlling the project progress
Explain the mechanism used for software configuration management and bug/feature tracking
Develop the communication, leadership and teamwork skills necessary to effectively work as professionals in teams, or in charge of teams

VII. Assessment Strategy

Students are assessed based on: (a) their performance in two exams (midterm and final); (b) their quizzes; and (c) the quality of a final team project and related oral presentation. The relative weights of each of these items on the final grade are as follows:
There will be one midterm exam accounting for 20% and final exams account for 30% of the grade. Combined, they account for 50% of the grade.
Class quizzes account for 20% of the grade.
Class participation accounts for 5% of the grade, and is evaluated based on the student's active participation in speakers' presentations, and the ability of students to add to the material already provided by the instructor to them
The class project accounts for 25% of the grade. It is evaluated based on a project document, oral presentation, and client perceptions of the team project. The project must be conducted in collaboration with a client organization (for example, a department at a large company or non-profit organization). A letter from the main contact person at the client organization, discussing and evaluating the project and its outcomes, must be provided to the instructor. The letter should contain the contact information of the person writing so the instructor can call him/her up and inquire about the project.

The exams encourage the students to review all of the concepts and methods discussed in class, which are primarily based on textbook material. This is complemented by the class discussions on recent articles taken from online industry publications, which allow the students to become conversant with the industry-specific lingo related to software engineering issues. The final project provides an experience where concepts, methods, and industry-relevant issues are all brought together in a very applied manner to solve a real problem faced by a real organization. While this project is not as extensive as a program capstone project, it gives the students the necessary exposure to industry-relevant issues to prepare them for the future challenge of conducting a final program capstone project, and subsequently pursuing a successful career as IT professionals.

Evaluation	
Class Participation	5%
Project	25 %
Quizzes	20%
Midterm Exam	20 %
Final Exam	30 %
Total	100%

Grading Scale

A+	96 - 100%
A	90 - 95 %
B+	86 – 89 %
B	80 – 85 %
C+	76 – 79 %
C	70 – 75 %
D+	66 – 69 %
D	60 – 65 %
F	Below 60 %

VIII. Course Format

This course utilizes a mix of in-class lectures, discussions, and software demonstrations designed to help students learn the various tasks involved in managing, designing and implementing a large scale software project. While class meetings are utilized to emphasize conceptual foundation in topics related to these tasks, software demonstrations are used to provide students with hands-on training in performing these tasks. Students are expected to attend three hours of lecture per week. There are no scheduled lab hours for this course.

In addition, the instructor should consider creating a Web site for this course using Web technologies such as WebCT or BLACKBOARD. At minimum, the site should include:
Course syllabus
Lecture material (for example PowerPoint slides, lecture notes, etc.). These should be placed on the site ahead of class meeting so that students may use the material to prepare for the lecture
Software demonstration exercises completed in class
Out-of-Class assignments and end-of-term project
Mechanism for students to digitally submit their assignments
Course calendar
Mechanism to communicate electronically (for example e-mail)
Discussion groups
Students course performance measures

Classroom Hours (3 hours per week) Class: 3

Lab: 0

IX. Topics to be covered

Week	Chapters	Topics
1	Introduction/ Chapter1	What Is a Project? What is Project Management?
2-3	Chapter 2	Understanding The Project Management Process Groups
4	Chapter 3	How to Scope a Project
5-6	Chapter 4	How to Plan a Project/ Introduction to Redmine
7-8	Software Configuration Management Tool	Introduction to Subversion (Midterm Exam)
9	Bug/Feature Tracking	Introduction to Bug/Feature Tracking
10	Chapter 5	How to Launch a Project
11	Chapter 6	How to Monitor and Control a Project
12	Chapter 8	Project Management Landscape
13	Chapter 7	How to Close a Project
14-15	Project/Review	Project Presentations, Course Review/Advanced Topics
16	Final Exam	(Final exam. Scheduled by the Registrar)

XI. Laboratory Exercises

There is no lab component for this course. However, students are assigned one out-of-class application development exercise every three weeks. These hands-on exercises are expected to be done in a team setting (generally 3-4 students/team) and are designed to illustrate various development concepts covered during lecture meetings. Thus, students are expected to complete about four such exercises. Collectively, these exercises enable students to learn the several steps involved in the scoping, planning, launching and monitoring the project using an online project management tool (e.g. Redmine). The following major areas should be covered in these exercises:
Identify project requirements
Develop RBS and WBS of the requirements
Develop software design documents (class diagram, ER diagram)
Develop a plan for the project
Monitor project progress and obtain weekly status reports

XI. Technology Component

A. In class, the instructor makes use of state-of-the art multimedia projection equipment and software. These are used to project slides and Web-based content relevant to the concepts of software project management
B. In class, in a laboratory setting, the instructor makes use of a open source project management software to demonstrate various tasks involved in successful design and implementation of software systems.

C. Outside class, the instructor uses Web-based course management software (for example WebCT, BLACKBOARD) to interact with students as described under course format section.

XII. Special Projects/Activities

An end-of-term team project is required in this course. This project is designed to permit students to apply concepts, methods, and tools learned in class to manage software system for a fictitious company. The projects require students, working in teams, to scope, plan, design and implement a complete software system. Student teams are expected to:
Gather and document requirements
Develop project plan and project schedule
Design, test, and deploy the application,
Make an oral presentation of their solution.

The final project is thus designed to assess competency in performing various tasks related to the software project management and its implementation.

XIII. Textbooks and Teaching Aids

A. Required Textbook

Effective Project Management: Traditional, Agile, Extreme, 5th Edition
Robert K. Wysocki ISBN: 978-0-470-42367-7 April 2009

B. Alternative Textbooks

C. Supplemental Print Materials
INFORMATION TECHNOLOGY PROJECT MANAGEMENT, 6th Edition Kathy Schwalbe
Managing and Leading Software Projects Richard E. (Dick) Fairley ISBN: 978-0-470-29455-0, March 2009

D. Supplemental Online Materials

As available from publisher.

Class RULES

- Disruptive Behavior
  Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
- Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance

Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non-attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness

When a student is late (5 minutes) for 3 times it is counted as one absent.

WITHDRAWAL

It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W", after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

ITAP 3371: Database II

Prerequisites

GEIT 3341: Database I

Credit hours

3 Credit Hours

Course overview

This course is a continuation of GEIT 3341 (Database I) and covers more advanced topics in database systems including advanced SQL, query processing & optimization, transaction processing, concurrency control, database recovery, database security & authorization, object-relational databases, and distributed databases.

PMU Competencies and Learning outcomes

This is an advanced course in database systems. It aims at broadening students' database knowledge by covering more advanced topics. In addition to the basic and intermediate SQL acquired in Database I, the course extends students' SQL skills by covering more advanced features in SQL. Moreover, the course provides the students with an understanding of how queries are processed and optimized within the database management system. To help the students understand how transactions are processed in a multi-user environment, relevant topics including transaction processing, concurrency control, and recovery techniques are covered. Furthermore, students learn how to protect the security of a database against unauthorized access, create and manipulate object-relational and distributed databases. Finally, the course provides the students with the communication, leadership and teamwork skills necessary to effectively work as professionals in teams, or in charge of teams, responsible for developing database application programs.

Detailed course description

The course begins by covering advanced SQL features including interaction with Java programs, simple and searched case, rank, running totals, percent to total, and sequences. Following this is a discussion of query processing and optimization where the steps involved in query processing, and how heuristics are used in query optimization, are covered. Topics related to transaction processing are then covered including transaction and system concepts, desirable properties of transactions, schedule types, and characterizing schedules based on serializability. Lock-based concurrency control techniques are discussed next including binary, shared/exclusive, and two-phase. Following this is a discussion of recovery techniques in single and multi-user settings. Next, are topics related to database security and authorization including types of security, discretionary access control based on granting and revoking of privileges, and types of security issues. Object-relational concepts are covered next including user-defined types and functions, nested tables and Varrays. Finally, the course concludes with a discussion of distributed databases including data fragmentation, replication and allocation, and types of distributed database systems.

Learning Outcomes

In this course, students learn:
Describe, Contrast and Apply advanced DDL and DML SQL
Describe and Contrast query processing steps and optimization techniques
Describe transaction processing and concurrency control techniques
Describe and Contrast database security and recovery techniques
Describe, Contrast and Apply object-relational databases
Describe and Contrast distributed database concepts and techniques
Download, Install and Configure a professional database system, e.g., Oracle or MySQL
Develop the communication, leadership and teamwork skills necessary to work in, or in charge of teams, to develop database applications in a high level programming language like Java

Required Text

Required Textbook

Ramez Elmasri and Shamkant Navathe , Fundamentals of Database Systems, Pearson; 7 edition (June 18, 2015)
Language: English
ISBN-10: 0133970779
ISBN-13: 978-0133970777

For the lab component of the course

Steven Feuerstein; Oracle PL/SQL Programming; 6th edition (February 16, 2014); O'Reilly Media
ISBN-10: 1449324452

class RULES

Disruptive Behavior
Any true discussion or application of hands-on laboratory assignments involves personal exposure and thus the taking of risks. Your ideas and application may not jibe with your neighbor's yet as long as your points are honest and supportable; they need to be respected by all of us in the classroom. Encouragement, questions, discussion, and laughter are a part of this class, but scoffing is never allowable, just as disruptive behavior is grounds for dismissal.
Use of Mobile phones, Blackberry, Ipods, etc is strictly prohibited during class.

Class attendance

Class attendance is required and a class roll will be taken during each class period. Learning is an active process, and it is simply impossible for you to participate if you aren't here. Your participatory attendance is important to achieving the learning outcomes. If non-attendance occurs you will be responsible for materials covered during your absences, and it is your responsibility to consult with me. Please refer to the university wide policy for class attendance in your Student Guide handbook. According to PMU Policy a student will receive 5%, 10% and 15% warnings for number of absences. The student will be withdrawn from the class if he/she exceeds 15% absences without excuse and 25% with or without excuse. The excuses are submitted to the instructor and are approved or rejected by the instructor.

Tardiness

When a student is late for 3 times it is counted as one absent.

WITHDRAWAL

It is the responsibility of the student to officially drop or withdraw from a course. Failure to officially withdraw may result in the student receiving a grade of "F" in the course. If you decide to drop the course, please discuss this with your instructor before taking this step. The student has the first ten weeks of the semester to withdraw with a grade of "W" , after that it will depend on the status in the course and the student can receive a grade of "WF" or "WP".

Assessment

Students are assessed based on: their performance in two exams (midterm and final); their class participation, which includes programming assignment, discussion of recent articles taken from online industry publications; and the quality of a final team project and related oral presentation. The relative weights of each of these items on the final grade are as follows:

GRADING SCHEME:

Participation	5%
Quizzes	5%
Lab	
Individual Labs:	10%
Midterm Lab Exam:	5%
Final Lab Exam:	5%
Midterm	20%
Project	20%
Final	30%
Total	100%

Course Format

A. Instruction

Four of the course's class meetings are used for laboratory demonstrations and activities geared at helping the students learn the several steps involved in designing and implementing a database system. The other class meetings are split into two main components: lectures, and class discussions. The lectures cover several topics outlined later in this syllabus. The class discussions are based on recent articles taken from online industry publications such as the Searchers and CIO magazines, which are freely available from the Web. The instructor provides the links to the articles, which are then downloaded by the students and read prior to class. In class, the students discuss the articles in small teams for about 20 minutes, developing three provocative questions per team. This is followed by a discussion involving the whole class, where each team asks one of the questions they developed, and other teams answer them, until all teams asked at least one of their questions. This discussion format is likely to lead to lively debate on topics that are directly addressed by the article, as well as on topics that are indirectly related to the article.

B. Web supplement

Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:
Course syllabus
Course assignments
Sample solutions to examinations (after being graded and returned)
Sample solutions to programming assignments (after being graded and returned)
Course calendar (an active utility)
Course e-mail (an active utility)
Course discussion list (an active utility)
Student course performance (an active utility)

CLASS SCHEDULE (Tentative)

Week	Chapters	Topics
1,2	Chapter 9	Advanced SQL
3,4	Chapter 22	Object-Relational and Extended-Relational Systems
5,6	Chapter 15	Query Processing Query Optimization
7,8,9	Chapter 17	Transaction Processing: Concepts & Theory
10,11	Chapter 18	Concurrency Control Techniques
12	Chapter 19	Database Recovery Techniques
13	Chapter 23	Database Security
14	Chapter 25	Distributed Databases and Client-Server Architectures
15	REVISION	

Project Samples

Sample #	Title
1	Music Sales Database Application
2	Inventory Control Management Database Application
3	Hospital Management Database Application
4	Library Management Database Application
5	Payroll Management Database Application

Lab Plan

Lab #	Title
0	PreLab
1	Advanced SQL – Simple & Searched CASE
2	Advanced SQL – Ranking
3	Advanced SQL – Sequences
4	Advanced SQL – Running Totals
5	Advanced SQL – Percent to Total
6	Java Oracle Interaction
7	Object-Relational: User Defined Types & Functions
8	Object-Relational: Nested Tables
9	Object-Relational: Varrays
10	IMP & EXP Utility in Oracle

F. COMPUTER ENGINEERING COURSES

- COEN 2411: Circuits
- COEN 3323: Digital and Logic Design
- COEN 3421: Electronics
- COEN 3361: Computer Networks
- COEN 4322: Digital Signal Processing
- COEN 4361: Operating Systems
- COEN 4413: Embedded Systems

COEN 2411: Circuits

Semester Credit Hours: 4 (3,2,4)

I. Course Overview

This course covers important theory in DC and AC circuits analysis. Topics include a review of the solution of simultaneous equations; Kirchoff's Current and Voltage Laws; nodal and mesh circuit analysis; superposition; source transformations; Thevenin and Norton Equivalent circuits; ideal op-amps; and RC, RL, and RLC circuits.

II. PMU Competencies and Learning Outcomes

Skills in understanding of DC and AC circuit theory are major components of professional competence for electrical and computer engineers. Throughout the semester, students are encouraged to apply critical thinking and problem solving skills in the class discussions and assignments. Professional communication skills (written and oral) are encouraged through discussions and assignments. Effective use of the most modern technology is integral to the development of the knowledge and skills acquired in this class.

III. Detailed Course Description

This course covers important theory in DC and AC circuits analysis. Topics include a review of the solution of simultaneous equations; Kirchoff's Current and Voltage Laws; nodal and mesh circuit analysis; superposition; source transformations; Thevenin and Norton Equivalent circuits; ideal op-amps; and RC, RL, and RLC circuits.

IV. Requirements Fulfilled

This is a required course for majors in computer engineering.

V. Required Prerequisites

Successful completion of:

- MATH 1324: Calculus III
- PHYS 1422: Physics for Engineers II

VI. Learning Outcomes

At the end of this course, students will:

- Be able to accurately define current, voltage, energy, and power in DC and AC circuits.
- Be able to solve for current, voltage, stored energy, and power in DC and AC circuits using the following techniques: Kirchoff's current and voltage laws; node voltage analysis; mesh current

analysis; superposition; and source transformations.

- Be able to define time constants for RC and RL circuits.
- Be able to calculate the total response of RC, RL, and RLC circuits.

In addition to these outcomes, students develop an intuitive understanding of DC and AC circuits such that they can use this intuition in the analysis and design of circuits

VII. Assessment Strategy

The assessment strategy measures students' understanding of circuit theory and apply the knowledge acquired in the analysis and design. This is achieved in the following ways:

- Class participation is used to help indicate to the instructor and the student his or her level of involvement and understanding.
- Homework assignments are used to provide feedback to students and to indicate individual progress in meeting course goals.
- Mid-term examinations are used to indicate students' developing level of mastery of the topics of the course.
- An end-of-semester final examination is used to measure the student's mastery in understanding and application of the knowledge integral to the course.

Assessment in this course is designed to assist students to further their understanding of the university's learning objectives. In cooperation with the instructor, each student selects two design assignments to become a part of the student's portfolio.

VIII. Course Format

The course consists of lectures, class discussions, homework assignments to be completed outside of class, and examinations. Students prepare for class by reading the text and additional materials and by completion of assignments so that they may be discussed in class are expected as indicators of students' commitment to professional growth. In addition, a recitation session is used to give students practice and supervised instruction in the analysis of DC and AC circuits.

Classroom Hours: Class lecture: 3 hours per week
Recitation Session: 1 hour per week

IX. Topics to be Covered

A. Definition of physical quantities: current, voltage, energy, and power
B. Basic circuit components
C. Voltage and current laws
D. Nodal and mesh analysis
E. Superposition and source transformation
F. Thevenin and Norton equivalent circuits
G. Operational amplifiers
H. Capacitors and inductors
I. RL and RC circuits
J. RLC circuits

X. Laboratory Exercises

XI. Technology Component

Students in this class are expected to have a computer account with the appropriate server to enable class communications. Media assisted instruction is a tool in this class. Use of appropriate technology for analysis of data and completion of problems is required, for example, use of a scientific calculator,

and use of student owned laptop. Students utilize the application software packages (MATLAB and PSPICE) in homework problems.

Web supplement: The course homepage on the University's BLACKBOARD system includes the following.

- Course syllabus
- Course assignments
- Course e-mail utility
- Course discussion list
- Student course grades

XII. Special Projects/Activities

A student project is not required for this class.

XIII. Textbooks and Teaching Aids

A. Required Textbook

Sadiku, Matthew, and Charles Alexander, Fundamentals of Electric Circuits, 5th Edition, McGraw Hill, 2012.
ISBN 978-0073380575

B. Alternative Textbooks

None.

C. Supplemental Textbooks

O'Malley, John, Schaum's Outline of Basic Circuit Analysis, 2nd Edition, McGraw-Hill, 1992.

D. Supplemental Materials

1. Scientific calculator
2. Laptop computer
3. MATLAB® and PSPICE® access either on laptop or in a general purpose computer lab
4. Engineering paper
5. CRC Standard Mathematical Tables and Formulae, Daniel Zwillinger

COEN 3323: Digital and Logic Design

Semester Credit Hours: 3 (2,3,3)

I. Course Overview

This course addresses the understanding and design of digital systems. Topics progress through Boolean algebra and logic gates; combinational logic; sequential logic and synchronous sequential logic systems; and design of logic circuits.

II. PMU Competencies and Learning Outcomes

Knowledge of digital systems and skill in their design, as taught in this course, are major components world wide of professional engineering practice. Throughout the semester, students are assisted to develop this knowledge and skill. Students are encouraged in development of professional engineering competencies including critical thinking skills, problem solving skills, and application of these in class discussions, assignments, and lab exercises. Professional demeanor and a team approach to understanding problems are practiced throughout lectures and discussions. Professional active communication skills (written and oral) are encouraged through discussions and assignments. Students are led to develop awareness of the professional role and responsibilities of engineers in a global society. Effective use of the most modern technology is integral to the development of the knowledge and skills acquired in this course.

III. Detailed Course Description

This course presents students with knowledge and design applications in the field of Digital Systems. Students are led from the basics of Boolean algebra and logic gates through increasing understanding to the design of logic circuits.

IV. Requirements Fulfilled

This is a required course for majors in computer engineering

V. Required Prerequisites

Successful completion of:

- COEN 2411: Circuits

VI. Learning Outcomes

In this course, students:

- Acquire the ability to formulate and solve problems involving Boolean algebra.
- Learn to design digital systems using simple logic elements.
- Learn to apply Karnaugh Maps to digital logic systems.
- Develop understanding of digital codes and number systems.
- Develop understanding of sequential logic circuits and their applications.

VII. Assessment Strategy

The assessment strategy measures a student's understanding of digital systems and their design.

- Class participation is monitored as an indicator of each student's level of involvement, understanding, and commitment.
- Homework and lab assignments are utilized to provide feedback to students and to indicate individual progress in achievement of understanding.

- A student project and report are required as measures of the student's ability to integrate knowledge acquired and apply it in real-world examples.
- Communication skills are measured through the student's in-class participation in discussions, written assignments, and presentation of the student report.
- Examinations are used to indicate student's progress in mastery of course content and lab expertise.
- An end-of-semester final examination is used to measure the student's mastery in understanding and application of the knowledge and design skills in the course.

Assessment in this course is designed to assist students to further their understanding of the university's learning objectives. Student's preparation for the capstone experience is enhanced through progressive skill building in active listening, oral and written communication, decision making individually and as a team member, problem solving, and professional viewpoint. In cooperation with the instructor, each student selects one assignment to become a part of the student's professional portfolio.

VIII. Course Format

The class consists of lectures, class discussions, written assignments to be completed outside of class, lab assignments, a student project and report, and examinations.

Classroom Hours: Class: 2 hours per week classroom lecture
Lab: 3 hours per week

IX. Topics to be Covered

A. Introduction to digital concepts
B. Number systems
   1. Operations
   2. Codes
C. Logic gates
D. Boolean Algebra and logic simplification
E. Karnaugh Maps
F. Combinational logic
G. Sequential logic circuits
H. Memory and storage
I. Introduction to microprocessors
J. Integrated circuit technologies

X. Laboratory Exercises

Weekly lab exercises supplement instruction in the classroom and provide each student with hands-on utilization of digital system hardware and measurement instrumentation. Weekly labs are performed on topics as follows:

A. Instruments and measurements
B. Logic gates and Boolean laws
C. DeMorgan's Theorems
D. Combinational logic circuits
E. Universal property of NAND and NOR gates
F. Adders and multiplexeres
G. Encoders and decoders
H. Seven-segment display
I. Comparators
J. Look-ahead carry adders
K. Arithmetic logic unit
L. Latches and flip-flops

M. Counters
N. Shift registers

XI. Technology Component

A. Students in this class are expected to have a computer account with the appropriate server to enable class communications. Media assisted instruction is a tool in this class. Use of appropriate technology for assignments and in-class work is required, for example, use of a scientific calculator, and use of the university computer labs. Use of the Internet may be indicated as notified by the instructor to support global understanding of applications.

B. Lab work for this course is completed using the lab exercises and appropriate technology in the PMU Digital Systems and Computer Architecture Laboratory. The lab experience is designed to integrate knowledge and theory into applied practice.

C. Web supplement

The course homepage on the University's BLACKBOARD system includes the following.

- Course syllabus
- Course assignments
- Course e-mail utility
- Course discussion list
- Student course grades

XII. Special Projects/Activities

Students complete a project and present a project report (written and oral presentation). The project should demonstrate the student's ability to utilize the knowledge acquired in an application of professional quality.

XIII. Textbooks and Teaching Aids

A. Required Textbook

Floyd, Thomas L., Digital Fundamentals, 11th Edition, Prentice Hall, 2014.
ISBN 978-0132737968

B. Alternative Textbooks

Widmer, Neal S., Gregory L. Moss, and Ronald J. Tocci, Digital Systems: Principles and Applications, 12th Edition, Pearson Prentice Hall, 2016. ISBN 0134220137

Tocci, Ronald J., Neal S. Widmer, and Gregory L. Moss, Digital Systems: Principles and Applications, 9th Edition, Pearson Prentice Hall, 2004.
ISBN 0-13-111120-5

Balabanian, Norman, and Bradley Carlson, Digital Logic Design Principles, John Wiley & Sons, 2001.
ISBN: 0-471-29351-2

C. Supplemental Print Materials

As notified by the instructor.

D. Supplemental Online Materials

As notified by the instructor.

COEN 3421: Electronics

Semester Credit Hours: 4 (3,3,4)

I. Course Overview

This course is the first of two courses in the use of electronic devices in analog and digital circuits. The lecture component covers device physics and modeling of op-amps, diodes, FETs, and BJTs; single and multi-stage amplifiers; differential amplifiers; feedback; frequency response; Bode plots. Laboratory component covers generation and acquisition of signals; current, voltage, and impedance measurements; transfer function measurement; and spectrum measurements and analysis.

II. PMU Competencies and Learning Outcomes

Skills in analyzing and designing analog and digital circuits are major components of professional competence for electrical and computer engineers. Throughout the semester, students are encouraged to apply critical thinking and problem solving skills in the class discussions, assignments, and lab activities. Professional communication skills (written and oral) are encouraged through discussions and assignments. Effective use of the most modern technology is integral to the development of the knowledge and skills acquired in this class.

III. Detailed Course Description

This course is the first of two courses in the use of electronic devices in analog and digital circuits. The lecture component covers device physics and modeling of op-amps, diodes, FETs, and BJTs; single and multi-stage amplifiers; differential amplifiers; feedback; frequency response; Bode plots. Laboratory component covers generation and acquisition of signals; current, voltage, and impedance measurements; transfer function measurement; and spectrum measurements and analysis.

IV. Requirements Fulfilled

This is a required course for all computer engineering majors.

V. Required Prerequisites

Successful completion of:

- COEN 2411: Circuits

VI. Learning Outcomes

At the end of this course, students will:

- Be able to accurately define current, voltage, and power gain in amplifiers.
- Be able to accurate calculate the current-voltage characteristics of diode, FETs, and BJTs.
- Be able to define the small-signal characteristics of FETs and BJTs.
- Be able to use small-signal circuit models of FETs and BJTs in the analysis of circuits.
- Be able to calculate the frequency response of circuits.
- Be able to analyze circuits using diodes, FETs, and BJTs using modern electronics instrumentation.

VII. Assessment Strategy

The assessment strategy measures the student's understanding of electronics and applies the knowledge acquired in the analysis and design of circuits. This is achieved in the following ways:

- Class participation is used to help indicate to the instructor and the student his or her level of involvement and understanding.

- Homework assignments are used to provide feedback to students and to indicate individual progress in meeting course goals.
- Design problems – conducted both experimentally in the lab and theoretically using PSPICE® – are used to provide students with advanced design and analysis problems, done either individually or in groups, that focus on advanced critical thinking and problem solving skills.
- Mid-term examinations are used to indicate students' developing level of mastery of the topics of the course.
- An end-of-semester final examination is used to measure the student's mastery in understanding and application of the knowledge integral to the course.
- An end-of-semester lab practical exam is used to measure the student's mastery in understanding and application of the knowledge integral to the course.

Assessment in this course is designed to assist students to further their understanding of the university's learning objectives. In cooperation with the instructor, each student selects a design problem to become a part of the student's portfolio. In addition, each student keeps an engineering notebook that accurately reflects all activities done in the lab portion of this course.

VIII. Course Format

The course consists of lectures, class discussions, homework assignments to be completed outside of class, laboratory exercises and projects, and examinations. Students prepare for class by reading the text and additional materials and by completion of assignments so that they may be discussed in class are expected as indicators of the student's commitment to professional growth.

Classroom Hours: Class: 3 hours per week
Lab: 3 hours per wee

IX. Topics to be Covered

A. Introduction to electronics: signals and amplifiers
B. Operational amplifiers
C. Diodes
D. MOS field-effect transistors (MOSFETs)
E. Bipolar junction transistors (BJTs)
F. Single-stage amplifiers
G. Differential and multistage amplifiers

X. Laboratory Exercises

All laboratory exercises are designed to provide students with expertise needed to make measurements from analog and digital circuits using diodes, FETs, and BJTs. In addition a problem is assigned to focus on the design, implementation, and analysis of circuits to electronic applications.

XI. Technology Component

Students in this class are expected to have a computer account with the appropriate server to enable class communications. Media assisted instruction is a tool in this class. Use of appropriate technology for analysis of data and completion of problems is required, for example, use of a scientific calculator, and use of student owned laptop. Students utilize the application software packages (MATLAB and PSPICE) in homework problems and in labs.

Web supplement: The course homepage on the University's BLACKBOARD system includes the following.

- Course syllabus
- Course assignments
- Course e-mail utility
- Course discussion list
- Student course grades

XII. Special Projects/Activities

A student project is not required for this class.

XIII. Textbooks and Teaching Aids

A. Required Textbook

Adel Sedra and Kenneth Smith, Microelectronic Circuits, 7th Edition, Oxford University Press, 2014
ISBN: 978-0199339136

Boylestad, Robert, and Louis Nashelsky, Electronic Devices and Circuit Theory, 11th Edition, Prentice Hall, 2012.
ISBN 978-0132622264

B. Alternative Textbooks

None.

C. Supplemental Print Materials

1. Scientific calculator
2. Laptop computer
3. MATLAB and PSPICE access either on laptop or in a general purpose computer lab
4. Engineering notebook

C. Supplemental Online Materials

None.

COEN 3361: Computer Networks

Semester Credit Hours: 3 (3,0,3)

I. Course Overview

This course covers the architecture and protocols of local and wide area networks, including signaling, data representation, error control, flow control and routing. The Java-based interprocess communication API is used to provide practical examples of communication, error control, and flow control. Peer to peer and client/server configurations based upon Unix/Linux and Windows architectures are explored.

II. PMU Competencies and Learning Outcomes

This course concentrates on theoretical and technical issues. Students in this course enhance their interpersonal and group effectiveness skills.

III. Detailed Course Description

COEN 3361: Computer Networks is concerned with the structure of data communications; from the electric interface, flow control, medium access control protocols, through data transmission and network protocols, packet switching and frame relay protocols, and includes an examination of network standards, and open systems.

IV. Requirements Fulfilled

This course is a required course for students in the Computer Engineering program and an available elective for Computer Science and Information Technology students in the College of Computer Engineering and Science. It should be taken no earlier than the senior year.

V. Required Prerequisites

MATH 2331 Linear Algebra

GEIT 3421 Data Structures

VI. Learning Outcomes

In this course, students learn:

- To develop an understanding of physical, theoretical and structural issues related to the construction of computer networks.
- To develop an understanding of the principles of electronic communication and the practical application to the organization and management of networks.
- To develop programming skills in developing interprocess communication applications.
- To develop improved communication and collaborative skills in meeting security threats as a team member or team leader.

VII. Assessment Strategy

This course is designed to introduce students to the physical, theoretical and practical principles underlying computer networks. Course grades are be based on:

- Weekly assigned homework to motivate students to do the work and earn credit accordingly.
- Weekly readings in current network literature.
- Four programming assignments related to specific networking issues.
- Two in-class exams to assess the student's accumulative mastery of content covered prior to time of exam.

- A comprehensive final exam to assess the student's accumulative mastery of course material.

The final grades is based on 20% credit for the homework, 20% for the weekly readings, 20% for the academic writing assignments, 20% on in-class exams, and 20% for the final examination.

VIII. Course Format

This course is primarily a lecture course. Students are expected to attend three hours of lecture per week.

Classroom Hours (3 hours per week) Class: 3

Web supplement: Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:

- Course syllabus
- Course assignments
- Keys to exams (after students have completed them)
- Course calendar (an active utility)
- Course e-mail (an active utility)
- Course discussion list (an active utility)
- Students course marks. (an active utility)

IX. Topics to be Covered

A. Open standards OSI/ TCP, ISO/RM
B. Electrical interface, transmission media
C. Data representations
D. Error control/ flow control
E. Data link protocols
F. Commercial grade transmission protocols
G. Packet switching and frame relay
H. Addressing, routing and congestion control
I. Data/voice and video transmission issues
J. Session and presentation protocols
K. Application protocols

X. Laboratory Exercises

There are no lab exercises for this course.

XI. Technology Component

This course makes use of the university's wireless access infrastructure during the class/ lecture sessions. In addition, the course makes use of the university's Blackboard interactive communication tool to enhance communication between the instructor and the students.

XII. Special Projects/Activities

There are four programming activities that are designed to develop a practical understanding of key data communication issues:

- Establishing interprocess communication between a client process and a server.
- Detecting communication errors resulting from garbled data.
- Detecting communication errors resulting from duplicate packets.
- Detecting communication errors resulting from missing packets.

XIII. Textbooks and Teaching Aids

A. Required Textbook
Tanenbaum, Andrew S., Computer Networks, 5th Edition, Prentice Hall, 2010.
ISBN 978-0132126953

B. Alternative Textbook
Stallings, William, Data and Computer Communications, 10th Edition, Prentice Hall, 2013.
ISBN 978-0133506488

C. Supplemental Print Materials

As available from publisher.

D. Supplemental Online Materials

As available from publisher.

COEN 4331: Microprocessors

Semester Credit Hours: 3 (2,3,3)

I. Course Overview

This course presents the development of microprocessor systems, with an introduction to assembly language programming. Instruction includes hardware-software interactions, programming techniques, and control of real-time hardware. Through the classes and labs, students are led to integrate knowledge into hands-on design and control applications.

II. PMU Competencies and Learning Outcomes

The ability to design, assemble, and test microprocessor systems, as taught in this course, is a major component world wide of professional engineering expertise. Throughout the semester, students are encouraged to develop and use critical thinking and problem solving skills as they work with hardware and software technologies. As students progress through class assignments, class discussions, lab exercises, and projects, students are encouraged in development of team leadership qualities and professional active communication skills. As they progress to design capabilities, students are led to develop growing awareness of the engineer's ethical role in a global society. Effective hands-on use of the most modern technology is integral to the development of the knowledge, skills, and professionalism acquired in this course.

III. Detailed Course Description

In this course, students learn the components of microprocessors and learn to design and assemble microprocessor systems with applications to real-world engineering environments. Instruction covers microprocessor architecture and assembly language programming, hardware-software interactions, programming techniques, and control of real-time hardware. Students are led to consider the leadership role and societal responsibilities inherent in a professional, ethical, engineering approach to use of microprocessor systems.

IV. Requirements Fulfilled

This is an ELECTIVE course for majors in electrical engineering.

V. Required Prerequisites

COEN 3323: Digital Logic Design
GEIT 3331: Computer Organization

VI. Learning Outcomes

- To learn the components of basic microcomputer architecture.
- To learn to program a microcomputer in assembly language.
- To develop the skill to design standard interfaces for microprocessors.
- To learn to contribute effectively as a member of an engineering team.
- To develop an understanding of ethical issues and the engineer's responsibility in society.

VII. Assessment Strategy

The assessment strategy measures students' understanding of microcomputer fundamentals, microprocessor architecture, assembly language, microcomputer interfacing. Through their various assignments and discussions, students are expected to demonstrate a growing understanding of awareness of themselves as professional engineers.

- Class participation is monitored as an indicator of each student's level of involvement, understanding, and commitment.
- Homework and lab assignments are utilized to provide feedback to students and to indicate individual progress in achievement of understanding.
- Student projects are required as measures of student's ability to integrate knowledge acquired and apply it in real-world examples.
- Computer programming skills are measured to indicate students' understanding of appropriate computer technologies and their applications in microprocessor systems.
- Communication skills are measured through the student's in-class participation in discussions, written assignments, and student projects.
- Examinations are used to indicate student's progress in mastery of course content and hands-on lab expertise.
- A final examination measures the student's mastery in understanding and application of the knowledge, the design skills, and the professionalism taught in the course.

Assessment in this course is designed to assist students to further their understanding of the university's learning objectives. The student's preparation for the capstone experience is enhanced through progressive skill building in active listening, oral and written communication, decision making individually and as a class team member, problem solving, use of appropriate technology, and professional viewpoint. In cooperation with the instructor, each student selects one assignment to become a part of the student's professional portfolio.

VIII. Course Format

The class consists of lectures, class discussions, and homework assignments including computer programming to be completed outside of class, student projects, lab exercises, and examinations. Preparation for class includes reading the text and additional resources and completing assignments so that the topics can be discussed in class. These are expected as indicators of students' commitment to professional growth.

Classroom Hours (5 hours per week) Class: 2

Lab: 3

IX. Topics to be Covered

A. Microcomputer fundamentals
   1. Number systems
   2. Codes
   3. Digital circuits
   4. Memory devices

B. Microprocessors
   1. Elements
   2. Structures
   3. Operation
   4. Memory
   5. Bus architecture
   6. Instruction set

C. Microcomputer programming
   1. Assembly language
   2. Arithmetic operations
   3. Decisions
   4. Loops
   5. Tables
   6. Lists
   7. Subroutines
   8. Interrupts

D. Microcomputer Interfacing
   1. Input/output modes
   2. Serial and parallel interfaces
   3. Synchronous and asynchronous communication

E. Development of a professional engineering viewpoint

X. Laboratory Exercises

Weekly lab exercises supplement instruction in the classroom and provide each student with hands-on utilization of microprocessor knowledge and skills.

XI. Technology Component

A. Computer Account

Students in this class are expected to have a computer account with the appropriate server to enable class communications. Media assisted instruction is a tool in this class. Use of appropriate technology for assignments and in-class work is required, for example, use of a scientific calculator or use of the university computer labs. A Motorola 68HC11 Development Board is required for student assignments. Use of the Internet may be indicated as notified by the instructor to support global awareness of applications.

B. Lab Work

Lab work for this course is completed using the lab exercises and appropriate technology in the PMU Digital Systems and Computer Architecture Laboratory. The lab experience is designed to integrate knowledge and theory into applied practice.

XII. Special Projects / Activities

Students complete a design and assembly project as a part of this course. Projects are expected to

demonstrate the student's ability to utilize knowledge acquired in an application of professional quality.

XIII. Textbooks and Teaching Aids

A. Required Textbook

Tocci, Ronald J., and Frank J. Ambrosio, Microprocessors and Microcomputers: Hardware and Software, 6th Edition, Prentice Hall, 2003.
ISBN 0130609048

B. Alternative Textbook

Miller, Gene H., Microcomputer Engineering, 3rd Edition, Pearson Prentice Hall, 2004.
ISBN 0-13-142804-7

C. Supplemental Print Materials
None.

D. Supplemental Online Materials
As notified by the instructor.

E. Other
Motorola 68HC11 Development Board.

COEN 4322: Digital Signal Processing

Semester Credit Hours: 3 (2,3,3)

I. Course Overview

The course presents an overview of the nature of signals, the algorithms and techniques used to process those signals and the applications to which digital signal processing can be usefully put.

II. PMU Competencies and Learning Outcomes

Students of COEN 4322: Digital Signal Processing develop the required mathematics to understand, implement and apply appropriate algorithms for signal processing as well as an understanding of the nature of signaling. This course is primarily theoretical and conceptual in nature with a strong practical implementation component. Students are expected to be able to design and implement solutions to signal processing problems and to communicate those solutions to their peers and to lay clients. Students are expected to work both individually and in groups to achieve a common objective. Students have the opportunity to enhance their adaptability, group effectiveness, and communication skills.

III. Detailed Course Description

COEN 4322: Digital Signal Processing is concerned with developing and understanding of the concepts underlying digital signal processing. The concept, structure, organization, and characteristics of signals are discussed with an examination of the spectrum of periodic signals and the frequency domain and the distinction between signal and noise, the causes of noise, and the effects of noise and other factors on signal quality. Techniques for processing signals are examined including filtering and non-filtering processes. Architecture and algorithms for signal processing are presented; graphical and spectral analysis, fast Fourier transforms, and the underlying concepts of

digital signal processors. Example applications for digital signal processing are presented including communication signal processing, speech signal processing, and sound signal processing.

IV. Requirements Fulfilled

This is a required course for all computer engineering majors. It may be used as an elective in the computer science degree program. This course should be taken in the senior year.

V. Required Prerequisites

- COEN 3323 Digital Logic Design
- MATH 3333 Probability and Statistics

VI. Learning Outcomes

In this course, students learn:

- To develop an understanding of the nature of signals and the limitations that media characteristics place on the utility of those signals.
- To develop expertise in the design and implementation of applications for the solution of digital signal processing problems.
- To become familiar with a variety of common DSP developmental tools.
- To be able to discuss the strengths and limitations of DSP applications in solving signaling problems with both professional peers and lay clients.
- To develop improved communication and group effectiveness skills.

VII. Assessment Strategy

The course grade involves an assessment of student performance on examinations that focus on the understanding of various concepts and constructs underlying Digital Signal Processing, and the communication of those concepts and the characteristics of designed solutions to DSP problems to an audience. Course grades are based on:

- Weekly assigned homework to motivate students to do the work and earn credit accordingly.
- Weekly, in-class presentations by students related to independent literature research on aspects of the course material and classroom discussion and critique of the presentation.
- Two in-class examinations to assess the student's accumulative mastery of content covered prior to the time of the examination.
- Three major programming assignments to test the student's understanding of the major concepts introduced during the course. Each programming assignment is assessed through instructor and peer review during in-class presentations.
- A comprehensive final examination to assess the student's accumulative mastery of course material.

The final grade is based on 10% credit for the homework, 10% for the presentations and participation in classroom discussion, 20% on in-class examinations, 40% on programming assignments, and 20% for the final examination.

Students are required to maintain a journal of thoughts and commentaries during the course. The journal contains daily entries including the identification of areas of interest and concern, notes on the preparation of presentations and comments, and analysis of classmate's presentations. The journal is reviewed weekly by the instructor to provide feedback to the students.

Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the

student's maturation as he proceeds through the curriculum.

VIII. Course Format

This course is a combination of lecture/discussion and laboratory use. Students are expected to attend two hours of lecture/discussion per week and to be available in a dedicated laboratory facility for three hours per week during which they should expect to undertake significant independent experimentation with DSP development tools. Once a week students should have at least 30 minutes of collaborative problem-solving activity.

Classroom Hours (5 hours per week)	Class: 2 Lab: 3

Web supplement: Course home page (Blackboard) should contain the following:

- Course syllabus
- Course assignments
- Sample solutions to examinations (after graded and returned)
- Course calendar (an active utility)
- Course e-mail (an active utility)
- Course discussion list (an active utility)
- Student course performance (an active utility)

IX. Topics to be Covered

A. Signal analysis
   1. Signals
   2. The spectrum of periodic signals
   3. Frequency domain
   4. Noise

B. Signal processing systems
   1. Systems
   2. Filters and non-filters
   3. Correlation
   4. Adaptation
   5. Biological signal processing

C. Architectures and algorithms
   1. Graphical techniques
   2. Spectral analysis
   3. Fast Fourier transform
   4. Digital filters
   5. Function evaluation algorithms
   6. Digital signal processors

D. Applications
   1. Communication signal processing
   2. Speech signal processing

X. Laboratory Exercises

There is a three-hour per week laboratory component to this course. The laboratory is not a structured time, but an opportunity for the students to work with DSP tools and hardware. There is no designated curriculum for laboratory exercises.

XI. Technology Component

This course makes use of the university's wireless access infrastructure. The course relies on the

university and the students having access to professional grade application development environments for the students to use. The course's laboratory component requires dedicated signal processing hardware and tools within one of the university-provided computer laboratories.

XII. Special Projects/Activities

Students are required to keep a "reflective notebook" in which, after each class, they enter their own assessments of what they learned, and what questions remain from the class. From each exercise set, each student selects one problem, which the student thinks best reflects the way the mathematical topic is used in a technical context. A detailed solution to the problem is included in the student's reflective notebook.

Students are expected to develop three DSP assignments during the semester that include software engineering design, implementation, and documentation. These assignments are group assignments determined through negotiation between the group and the instructor. The laboratory time provided in the course provides the majority of the time required for implementation with additional hours outside laboratory time available for the students on a group basis as required.

XIII. Textbooks and Teaching Aids

A. Required Textbook

Proakis, John G., and Dimitris Manolakis, Digital Signal Processing: Principles, Algorithms and Applications, 4th Edition, Prentice Hall, 2007.
ISBN 9780131873742

Sanjit K. Mitra, Digital Signal Processing: A Computer-based Approach, 4th Edition, McGraw-Hill, 2010.
ISBN: 007-125579-6

B. Alternative Textbooks

None.

C. Supplemental Print Materials

None.

D. Supplemental Online Materials

As available from publisher.

COEN 4361: Operating Systems

Semester Credit Hours: 3 (3,0)

I. Course Overview

This course is the study of the principles, purposes, and organization of operating systems. The goal is to prepare students an understanding of the theory as well as practices of the design and implementation of operating systems software.

II. PMU Competencies and Learning Outcomes

Students in this course develop conceptual and programming skills necessary for continued success in computer science. The skills enhance their abilities to appreciate the theory and practices of operating systems common to computer science as a discipline and to effectively communicate their solutions to fellow professionals. This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course includes individual as well as group projects, establishes both conceptual reasoning skills and technical communication skills, and provides opportunities for the presentation and defense of designed solutions.

III. Detailed Course Description

The Operating Systems course is concerned with the study of the principles, purposes, and organization of operating systems, including processes, tasks, scheduling, inter-process communication, synchronization, mutual exclusion, memory management, device management, file systems, security and protection, multi-CPU systems, computer networking, and distributed computing.

This course satisfies three hours of the requirements for the degree in computer engineering. It is required of all students pursuing a degree program in computer science and computer engineering within the College of Computer Engineering and Science. It should be taken in the first semester of the senior year.

IV. Requirements Fulfilled

This course is required for all students majoring in Computer Science and Computer Engineering in the College of Computer Engineering and Science. It is also recommended as an elective for students majoring in IT.

V. Required Prerequisites
GEIT 1412: Computer Science II
GEIT 2421: Data Structure
GEIT 3331: Computer Organization

VI. Learning Outcomes
In this course, students learn to:

1. Describe the underlying structure and services provided by operating systems
2. Describe and differentiate between the concepts of threads and processes
3. Explain and evaluate merits of different process scheduling algorithms
4. Explain inter process communication and process synchronization issues in cooperating processes or threads.
5. Explain the structure and design of memory and storage management in computer systems.
6. Demonstrate improved communication and collaborative skills.

VII. Assessment Strategy

This course is designed with three primary goals in mind: to introduce students to the conceptual basis and practical issues associated with the compiler construction, to enhance the student's programming techniques to its application in computer science, and to provide students with the opportunity to communicate their designs and implementations to their peers in a professional setting. With this in mind, the course grade involves an assessment of their performance on examinations that focus on the application of programming paradigms to the solutions of problems, the performance analysis of the designed solutions, and the communication of designed solutions to those problems to an audience. Course grades are based on:

- Homework to motivate students to do the work and earn credit accordingly.
- Major exams.
- Research Project
- A comprehensive final exam.

Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he/she proceeds through the curriculum.

VIII. Course Format

This course utilizes a mix of in-class lectures, discussions, and software demonstrations designed to help students learn the various tasks involved in designing and implementation of an operating system. While class meetings are utilized to emphasize conceptual foundation in topics related to these tasks, software demonstrations are used to provide students with hands-on training in performing these tasks. Students are expected to attend three hours of lecture per week. There are no scheduled lab hours for this course.

In addition, the instructor should consider creating a Web site for this course using Web technologies such as WebCT or BLACKBOARD. At minimum, the site should include:

- Course syllabus
- Lecture material (for example PowerPoint slides, lecture notes, etc.). These should be placed on the site ahead of class meeting so that students may use the material to prepare for the lecture
- Software demonstration exercises completed in class
- Out-of-Class assignments and projects
- Mechanism for students to digitally submit their assignments
- Course calendar
- Mechanism to communicate electronically (for example e-mail)
- Discussion groups
- Students course performance measures

Classroom Hours (3 hours per week) Class: 3
Lab: 0

IX. Topics to be covered

- I Virtual Machines (Vmware)
- Process Concept
- Multithread Programming
- Process Scheduling
- Synchronization
- Deadlock / Midterm Exam
- Memory Management
- Virtual Memory
- File Systems

- Systems Structures /Secondary Storage Structure

X. Laboratory Exercises

There is no lab component for this course. However, the students are assigned one out-of-class application development exercise every three weeks. These hands-on exercises are designed to illustrate various development concepts covered during lecture meetings. Thus, students are expected to complete about four such exercises. Collectively, these exercises enable students to learn the details of the operating systems. The following major areas should be covered in these exercises:

- Process and thread management
- Process synchronization
- Memory management
- File system management

XI. Technology Component

A. In class, the instructor makes use of state-of-the art multimedia projection equipment and software. These are used to project slides and Web-based content relevant to the concepts of operating systems.

B. Outside class, the instructor uses Web-based course management software (for example WebCT, BLACKBOARD) to interact with students as described under course format section.

XII. Special Projects/Activities

There will be multiple programming based mini-projects in this course. These projects are designed to permit students to apply concepts, methods, and tools learned in class. The students are expected to:

- Use operating system calls to create and manage processes
- Use operating system calls to create and manage thread
- Use system calls for memory and file system management

XIII. Textbooks and Teaching Aids

Silberschatz, A., P., B. Galvin, G. Gagne. Operating Systems Concepts: Wiley, Ninth Edition, 2012.

COEN 4413: Embedded Systems

Semester Credit Hours: 4 (3,2,4)

I. Course Overview

In this project-based course, students will design and develop an application for an embedded systems platform, and then investigate low-level performance tuning and optimization.

II. PMU Competencies and Learning Outcomes

Upon completion of this course, students should have developed their:

- Ability to identify, formulate, and solve engineering problems.
- Ability to communicate effectively.
- Ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.
- Ability to design a system, component, or process to meet desired needs with realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.

III. Detailed Course Description

This course incorporates topics from the domains of software engineering, compilers, operating systems, and computer architecture. It provides students with the foundation they will need for addressing the concerns of developing real-world embedded systems. Students are expected to be proficient in both C and Java.

IV. Requirements Fulfilled

This is a required course for all Computer Engineering majors.

V. Required Prerequisites

Successful completion of:

- GIET 2412: Data Structures
- GIET 3351 Software Engineering I
- COSC 4361: Operating Systems

VI. Learning Outcomes

Upon successful completion of this course, students should:

- Possess fundamental knowledge and skills in embedded system development methodologies, techniques and tools.
- Have the necessary educational foundation to apply various techniques for loop optimizations, data flow optimizations, instruction reordering, synchronization issues, and process scheduling.
- Have a good understanding and working knowledge for good software engineering and software design practices.
- Have an exposure to the Android API.
- Apply development methodologies, tools and techniques discussed in classroom in a real life group project.

VII. Assessment Strategy

Class performances	5%
Individual homework assignments	10%
Group project	40%
First major exam	15%
Final exam	30%

VIII. Course Format

The class consists of lectures, class discussions, and written assignments to be completed outside of class, examinations, and a student project. Students are expected to attend class and to participate in discussions and problem solving assignments. Students prepare for class by reading the text and additional resources and by completion of assignments so that they may be discussed in class are expected as indicators of students' commitment to professional growth.

Classroom Hours: Class: 3 hours per week
Project development: Time each week as needed outside of class

IX. Topics to be Covered

A. Introduction to Android programming (API, networking, location services, etc.); Weeks 1-4
B. Addressing real-world software concerns (performance, security, testing, etc.); Weeks 5-6
C. Working with virtual machines (compiler optimization, software pipelining, etc.); Weeks 7-9
D. Working with operating systems (scheduling, deadlock prevention, etc.); Weeks 10-11
E. Addressing real-world hardware concerns (power, memory usage, etc.); Weeks 12-13

X. Laboratory Exercises

This course does not require a separate lab.

XI. Technology Component

Students are expected to have a computer account with the appropriate server to enable class communications. Media assisted instruction is a tool in this class. Appropriate technology for analysis of data and completion of designs is required, for example, use of an engineering calculator, and use of the university computer labs. Completing assignments and examinations requires use of a personal computer and/or university computer labs. Use of the Internet may be indicated to support global understanding of applicability of skills.

Web supplement: The course homepage on the University's BLACKBOARD system includes the following.

- Course syllabus
- Course assignments
- Course e-mail utility
- Course discussion list
- Student course grades

XII. Special Projects/Activities

The course project will run for the duration of the semester, and students will work in groups of three to four. The project will consist of three "phases" in which students will explore the application layer, the virtual machine layer, and the operating system layer, as described below.

- In the first phase, students begin by identifying an application that they would like to build on the Android platform. The application should take advantage of the mobile computing platform

(e.g., use networking, location services, the embedded camera, etc.) but be somewhat simple in scale, in that development of the prototype should be complete within 4-5 weeks. The application will be developed on an Android emulator, as opposed to the actual hardware. This phase teaches students about good software engineering and software design practices, and exposes them to the Android API.

- In the second phase, students will investigate the underlying Java virtual machine (named Dalvik) that is part of the Android platform. By understanding how the virtual machine works, they will then be expected to make changes to both their application (developed in the first phase) and to Dalvik, using compiler techniques like loop optimizations, data flow optimizations, instruction reordering, etc.
- In the final phase, students consider the underlying operating system (a simplified version of Linux) and hardware. Students will learn about synchronization issues, process scheduling, etc. The course will use the Beagle Board embedded platform, which supports the open-source Android OS, so that students can consider physical issues such as energy drain, memory usage, etc
- XIII. Textbooks and Teaching Aids

A. Required Textbook

Barret, Steven F., and Daniel J. Pack, Embedded Systems: Design and Applications with the 68HC12 and HCS12, Prentice Hall, 2004.
ISBN 9780131401419

B. Alternative Textbooks

None.

C. Supplemental Print Materials

None.

C. Supplemental Online Materials

Assigned readings will be posted on the course home page, and a variety of supplemental material can be found under the Resources section on BLACKBOARD.

COEN 4331: Microprocessors

Semester Credit Hours: 3 (2,3,3)

I. Course Overview

This course presents the development of microprocessor systems, with an introduction to assembly language programming. Instruction includes hardware-software interactions, programming techniques, and control of real-time hardware. Through the classes and labs, students are led to integrate knowledge into hands-on design and control applications.

II. PMU Competencies and Learning Outcomes

The ability to design, assemble, and test microprocessor systems, as taught in this course, is a major component world wide of professional engineering expertise. Throughout the semester, students are encouraged to develop and use critical thinking and problem solving skills as they work with hardware and software technologies. As students progress through class assignments, class discussions, lab exercises, and projects, students are encouraged in development of team leadership qualities and professional active communication skills. As they progress to design capabilities, students are led to develop growing awareness of the engineer's ethical role in a global society. Effective hands-on

use of the most modern technology is integral to the development of the knowledge, skills, and professionalism acquired in this course.

III. Detailed Course Description

In this course, students learn the components of microprocessors and learn to design and assemble microprocessor systems with applications to real-world engineering environments. Instruction covers microprocessor architecture and assembly language programming, hardware-software interactions, programming techniques, and control of real-time hardware. Students are led to consider the leadership role and societal responsibilities inherent in a professional, ethical, engineering approach to use of microprocessor systems.

IV. Requirements Fulfilled
This is an ELECTIVE course for majors in electrical engineering.

V. Required Prerequisites
COEN 3323: Digital Logic Design
GEIT 3331: Computer Organization

VI. Learning Outcomes

- To learn the components of basic microcomputer architecture.
- To learn to program a microcomputer in assembly language.
- To develop the skill to design standard interfaces for microprocessors.
- To learn to contribute effectively as a member of an engineering team.
- To develop an understanding of ethical issues and the engineer's responsibility in society.

VII. Assessment Strategy

The assessment strategy measures students' understanding of microcomputer fundamentals, microprocessor architecture, assembly language, microcomputer interfacing. Through their various assignments and discussions, students are expected to demonstrate a growing understanding of awareness of themselves as professional engineers.

- Class participation is monitored as an indicator of each student's level of involvement, understanding, and commitment.
- Homework and lab assignments are utilized to provide feedback to students and to indicate individual progress in achievement of understanding.
- Student projects are required as measures of student's ability to integrate knowledge acquired and apply it in real-world examples.
- Computer programming skills are measured to indicate students' understanding of appropriate computer technologies and their applications in microprocessor systems.
- Communication skills are measured through the student's in-class participation in discussions, written assignments, and student projects.
- Examinations are used to indicate student's progress in mastery of course content and hands-on lab expertise.
- A final examination measures the student's mastery in understanding and application of the knowledge, the design skills, and the professionalism taught in the course.

Assessment in this course is designed to assist students to further their understanding of the university's learning objectives. The student's preparation for the capstone experience is enhanced

through progressive skill building in active listening, oral and written communication, decision making individually and as a class team member, problem solving, use of appropriate technology, and professional viewpoint. In cooperation with the instructor, each student selects one assignment to become a part of the student's professional portfolio.

VIII. Course Format

The class consists of lectures, class discussions, and homework assignments including computer programming to be completed outside of class, student projects, lab exercises, and examinations. Preparation for class includes reading the text and additional resources and completing assignments so that the topics can be discussed in class. These are expected as indicators of students' commitment to professional growth.

Classroom Hours (5 hours per week) Class: 2
Lab:

IX. Topics to be Covered

A. Microcomputer fundamentals
   1. Number systems
   2. Codes
   3. Digital circuits
   4. Memory devices

B. Microprocessors
   1. Elements
   2. Structures
   3. Operation
   4. Memory
   5. Bus architecture
   6. Instruction set

C. Microcomputer programming
   1. Assembly language
   2. Arithmetic operations
   3. Decisions
   4. Loops
   5. Tables
   6. Lists
   7. Subroutines
   8. Interrupts

D. Microcomputer Interfacing
   1. Input/output modes
   2. Serial and parallel interfaces
   3. Synchronous and asynchronous communication

E. Development of a professional engineering viewpoint

X. Laboratory Exercises

Weekly lab exercises supplement instruction in the classroom and provide each student with hands-on utilization of microprocessor knowledge and skills.

XI. Technology Component

A. <u>Computer Account</u>

Students in this class are expected to have a computer account with the appropriate server to enable class communications. Media assisted instruction is a tool in this class. Use of appropriate technology for assignments and in-class work is required, for example, use of a scientific calculator or use of the university computer labs. A Motorola 68HC11 Development

Board is required for student assignments. Use of the Internet may be indicated as notified by the instructor to support global awareness of applications.

B. Lab Work

Lab work for this course is completed using the lab exercises and appropriate technology in the PMU Digital Systems and Computer Architecture Laboratory. The lab experience is designed to integrate knowledge and theory into applied practice.

XII. Special Projects / Activities

Students complete a design and assembly project as a part of this course. Projects are expected to demonstrate the student's ability to utilize knowledge acquired in an application of professional quality.

XIII. Textbooks and Teaching Aids

A. Required Textbook

Tocci, Ronald J., and Frank J. Ambrosio, Microprocessors and Microcomputers: Hardware and Software, 6th Edition, Prentice Hall, 2003.
ISBN 0130609048

B. Alternative Textbook

Miller, Gene H., Microcomputer Engineering, 3rd Edition, Pearson Prentice Hall, 2004.
ISBN 0-13-142804-7

C. Supplemental Print Materials
None.

D. Supplemental Online Materials
As notified by the instructor.

E. Other
Motorola 68HC11 Development Board.