Course Title: COSC 4363: Automata Theory

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 Information Technology. 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 understand set theory and countability.
- To understand and apply finite automata and regular languages.
- To understand and apply push-down automata and context-free languages.
- 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.

The final grade 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 examinations, and 30% 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

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.