

Course Title: COSC 2331: Discrete Structures

Semester Credit Hours: 3 (3,0)

I. Course Overview

Discrete Structures 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.

II. PMU Competencies and Learning Outcomes

Students of COSC 2331: Discrete Structures 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.

III. Detailed Course Description

COSC 2331: Discrete Structures 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 logic and set operations, the analysis of algorithm complexity, mathematical reasoning including proofs and induction, recursion, and program correctness. Graphs, including Euler and Hamilton Paths and shortest path problems are examined, as are tree applications. The course concludes with an investigation of Boolean functions, gate representations and approaches to circuit minimization.

IV. Requirements Fulfilled

This course satisfies three hours of the requirements for the degree in computer science. It is required of all students pursuing the computer science degree from the College of Information Technology. It should be taken immediately after completion of MATH 1323: Calculus II.

V. Required Prerequisites

MATH 1423: Calculus II

VI. Learning Outcomes

In this course, students learn:

- To develop understanding of Logic Sets and Functions.
- To use mathematical reasoning techniques including induction and recursion
- To understand and apply counting techniques to the representation and characterization of relational concepts.
- To develop an understanding of how graph and tree concepts are used to solve problems arising in the computer science.
- To communicate the solutions of technical problems to other professionals.
- To develop improved collaborative skills.

VII. Assessment Strategy

This course is designed with three primary goals in mind: to further the student's understanding of mathematics, to lead students to connect the mathematics to its application in computer science 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 applications of discrete mathematics to computer science.

- 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 (the university's Web tool, WebCt or Blackboard) 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

A. Logic, sets, and functions

1. Logic
2. Propositional equivalences
3. Predicates and quantifiers
4. Sets and set operations
5. Functions

B. Algorithms, the integers, and matrices

1. Algorithms and their complexity
2. The integers and division
3. Integers and algorithms
4. Applications of number theory
5. Matrices

C. Mathematical reasoning

1. Methods of proof
2. Mathematical induction
3. Recursive algorithms
4. Program correctness

- D. Counting
 - 1. The pigeonhole principle
 - 2. Permutations and combinations
 - 3. Discrete probability
 - 4. Probability theory
 - 5. Generalized permutations and combinations
 - 6. Recurrence relations
 - 7. Generating functions
- E. Relations
 - 1. Relations and their properties
 - 2. N-ary relations and their applications
 - 3. Representing relations
 - 4. Closures of relations
 - 5. Equivalence relations
 - 6. Partial orderings
- F. Graphs
 - 1. Graph terminology
 - 2. Representing graphs and graph isomorphism
 - 3. Connectivity
 - 4. Euler and Hamilton paths
 - 5. Shortest path problems
 - 6. Planar graphs
 - 7. Graph coloring
- G. Trees
 - 1. Introduction to trees
 - 2. Applications of trees
 - 3. Tree traversal
 - 4. Trees and sorting
 - 5. Spanning trees
 - 6. Minimum spanning trees
- H. Boolean algebra
 - 1. Boolean functions
 - 2. Representing Boolean functions
 - 3. Logic gates
 - 4. Minimization of circuits

X. Laboratory Exercises

This course does not require a separate lab.

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

Rosen, K. H. (1999) *Discrete Mathematics*. McGraw-Hill.
ISBN 0-07-289905-0

B. Alternative Textbooks

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

C. Supplemental Print Materials

As available from publisher.

D. Supplemental Online Materials

As available from publisher.