CSC 230 Discrete Mathematical Structures for Computer Science

Prerequisites: *CSC 210 with grade of C or better; MATH 227 with grade of C or better or must be taken concurrently.*

Catalog Description: Review of set algebra, relations and functions; permutations; propositional logic; proof techniques; introduction to graph theory; infinite sets; applications to computer science.

Objective: The goal of this course is to introduce students to ideas and techniques from discrete mathematics that are widely used in Computer Science. We study topics in such areas as sets, logic, proof techniques, induction procedures, relations, functions, graphs, trees, combinatorics, and recursive procedures. All CSC majors should take this course.

Course Topics:

- **Sets, relations, and functions:** Sets (set operations, complements, Venn diagrams, membership table, Cartesian products, power sets, cardinality); Relations (reflexivity, symmetry, transitivity, compositions, closures, equivalence relations, equivalence class, partitions); Functions (injections, surjections, bijections, inverses, compositions);

- **Basic logic:** Propositional logic; Logical connectives; Truth tables; Normal forms (conjunctive and disjunctive); Validity; Predicate logic; Logical inference for deductive proofs; Modus ponens and modus tollens; Universal and existential quantification;

- **Proof techniques:** The structure of formal proofs; Direct proofs; Proof by Cases; Proof by counterexample; Proof by contraposition; Proof by contradiction; Mathematical induction; Strong induction; Recursive mathematical definitions; Deductive proof;

- **Basics of counting:** Counting arguments; Sum and product rule; Inclusion-exclusion principle; Arithmetic and geometric progressions; The pigeonhole principle; Set countability; Permutations and combinations; Pascal's identity; The binomial theorem; Fibonacci numbers;

- **Trees and Graphs:** Graphs; Graph operations; Graph types; Handshaking theorem; Graph representation (adjacency matrix and adjacency list); Isomorphism; Shortest path problem; Dijkstra’s algorithm; Trees; Spanning trees; Traversal strategies;

- **Introduction to Analysis of Algorithms:** Big Oh Notations; Solving recurrence relations; Computability theory (Halting problem); Number theory; Prime number; Modular arithmetic; Euclidean algorithm;

Learning Outcomes:
At the end of this course students will be able to

- Solve mathematical problems with discrete objects by identifying and using appropriate discrete mathematical methods
- Understand and create proofs of simple concrete problems
- Use correct logical inference to conduct deductive proofs
- Define and analyze functions
- Choose an appropriate method for counting discrete objects
- Solve the shortest-path problem by using Dijkstra's algorithm
- Analyze time complexity of an algorithm given in a pseudo code by deriving its big Oh

Method of Evaluation:
Student learning will be evaluated on the basis of

- Positive in-class participation
- Grade on homework
- Grade on midterm examinations
- Grade on final examination

The weight assigned to each element of evaluation will be determined by the instructor of the course on the first day of the class.

Missed Exams:
Generally, there will be no make-up exams and no incomplete grades given. If you miss an exam, you must notify the instructor before the exam or, if physically impossible, soon after. If any of the scheduled exam dates are in conflict with your religious observances, you must notify the instructor, in writing, at least two weeks in advance of the exam.

Students with disabilities:
Students with disabilities who need reasonable accommodations are encouraged to contact the instructor. The Disability Programs and Resource Center (DPRC) is available to facilitate the reasonable accommodations process. The DPRC is located in the Student Service Building and can be reached by telephone (voice/TTY 415-338-2472) or by email (dprc@sfsu.edu).

Attendance:
"Students are expected to attend classes regularly because classroom work is one of the necessary and important means of learning and of attaining the educational objectives of the institution." (SFSU Bulletin) To this end, attendance will be taken at different times through the term. Students missing class on a day of attendance will lose attendance points.

Recommended Textbooks
Discrete Mathematics for Computer Science, Haggard G, Schlipf J, Whitesides S.
Discrete Mathematics and Its Application, 6th Edition, Kenneth Rosen

Created and approved: November 2009

Revised: September 2012
Revised: November 2012 by Kazunori Okada
Revised: February 2015 by Kazunori Okada