Las Positas College
3000 Campus Hill Drive
Livermore, CA 94551-7650
(925) 424-1000
(925) 443-0742 (Fax)

Course Outline for CS 17

DISCRETE MATHEMATICAL STRUCTURES

Effective: Fall 2019

I. CATALOG DESCRIPTION:
CS 17 — DISCRETE MATHEMATICAL STRUCTURES — 4.00 units

Designed for majors in mathematics and computer science, this course provides an introduction to discrete mathematical structures used in Computer Science and their applications. Course content includes: Propositional and predicate logic; rules of inference; quantifiers; elements of integer number theory; set theory; methods of proof; induction; combinatorics and discrete probability; functions and relations; recursive definitions and recurrence relations; elements of graph theory and trees. Applications include: analysis of algorithms, Boolean algebras and digital logic circuits. Students who have completed, or are enrolled in, MATH 10 may not receive credit.

4.00 Units Lecture

Prerequisite
MATH 1 - Calculus I
with a minimum grade of C
(May be taken concurrently)

CS 1 - Computing Fundamentals I
with a minimum grade of C
(May be taken concurrently)

Grading Methods:
Letter Grade

Discipline:
- Mathematics or
- Computer Science

	MIN
Lecture Hours:	72.00
No Unit Value Lab	18.00
Total Hours:	90.00

II. NUMBER OF TIMES COURSE MAY BE TAKEN FOR CREDIT: 1

III. PREREQUISITE AND/OR ADVISORY SKILLS:
Before entering the course a student should be able to:

A. MATH1
   1. Evaluate the limit of a function at a real number;
   2. Determine whether a function is continuous at a point or an interval;
   3. Interpret the derivative as the slope of a tangent line and find the equation of a tangent line to a function;
   4. Evaluate the limit of a function at infinity;
B. CS1

IV. MEASURABLE OBJECTIVES:
Upon completion of this course, the student should be able to:

A. Construct truth tables;
B. Simplify and negate propositions and quantified predicates;
C. Use rules of inference to determine the validity of a logical argument;
D. Apply principles of propositional logic to Boolean algebras and simplification of digital logic circuits;
E. Use both direct and indirect proof techniques including proof by cases, proof by contraposition and proof by contradiction;
F. Write formal proofs using complete English sentences and principles of logic, rules of inference and quantification;
G. Use the Principle of Mathematical Induction to prove statements about sequences, series and algorithms;
H. Apply principles and definitions of set theory to find unions, intersections, set complement and prove statements about sets;
I. Solve counting problems and determine the probability of an event using elementary counting techniques;
J. Write a recurrence relation to model a sequence;
K. Represent a relation in both graphical and set forms; determine if a relation is an equivalence relation; find the equivalence classes of an equivalence relation;

L. Prove or disprove that a function is one-to-one or onto; find the composition of functions; find the inverse of function;
M. Demonstrate a knowledge of basic graph theory terminology; apply graph theory to shortest path problems;
N. Use properties of trees to solve sorting problems; find minimum spanning trees;
O. Apply the binomial theorem to independent events and Bayes' Theorem to dependent events;
P. Describe how formal tools of symbolic logic are used to model real-life situations, incuding those arising in computing contexts such as program correctness, database queries, and algorithms

V. CONTENT:

A. Propositional logic
   1. Logical connectives: Conjunction, disjunction, conditional, biconditional
   2. Logical equivalences
   3. Simplification and negation of propositions
   4. Truth Tables
   5. Rules of inference, including *Modus Ponens* and *Modus Tollens*
   6. Validity
B. Predicate logic
   1. Quantifiers, including universal and existential quantification; truth value of quantified predicates
   2. Negation
   3. Limitations of predicate logic
C. Proof Techniques
   1. Notions of implication, converse, inverse, contrapositive, negation and contradiction
   2. The structure of mathematical proofs
   3. Methods of proof
      a. Direct proof
      b. Proof by cases
      c. Proof by contraposition
      d. Proof by contradiction
      e. Counterexample
D. Induction
   1. Principle of mathematical induction
   2. Strong mathematical induction
   3. Well-ordering principle
   4. Recursive definitions
E. Elements of integer number theory
   1. Floor and ceiling functions
   2. Divisibility
   3. Quotient-Remainder Theorem
F. Sets and set operations
   1. Construction of sets: union, intersection, set diference, set complements, Cartesian product, power sets
   2. Venn diagrams
   3. Properties of sets
   4. Cardinality
G. Fundamentals of counting and discrete probability
   1. Counting arguments
   2. Sum and product rules
   3. Permutations and combinations
   4. Principle of inclusion/exclusion
   5. Pigeonhole principle
   6. Use of proof techniques in combinatorics
   7. Identities, including Pascal's formula
   8. Basic definitions and principles of finite probability:
      a. Finite probability space, probability measure, events
      b. Integer random variables, expectation
      c. Law of large numbers
      d. Finding the probability of an event
      e. The Binomial Theroem
      f. The Master Theorem
      g. Conditional probability, Bayes' Formula and independent events
H. Relations
   1. Representation of relations
      a. Set representation
      b. Graphical representation
   2. Closure
   3. Properties of relations: reflexive, symmetric, transitive, antisymmetric
   4. Equivalence relations and equivalence classes
I. Functions
   1. Definition of function
   2. One-to-one (injective) and onto (surjective) functions; bijections
   3. Composition of functions
   4. Inverse functions
J. Recursive definitions and recurrence relations
   1. Arithmetic and geometric sequences
   2. Fibonacci sequence
   3. Recursively defined sequences
K. Graph theory
   1. Definitions and representation of undirected and directed graphs
   2. Connectivity
   3. Euler and Hamilton paths
   4. Shortest path problems
   5. Planar graphs
L. Trees
   1. Definitions and applications
   2. Traversal and sorting
   3. Minimum spanning trees.
M. Applications
   1. Boolean algebras and digital logic circuits
      a. Normal forms (conjunctive and disjunctive)
      b. Simplification of circuits
   2. Euclidean algorithm; Division algorithm

VI. METHODS OF INSTRUCTION:

A. **Lecture** -
B. **Lab** - Problem solving laboratory including individual work and collaborative learning activities.
C. Writing assignments
D. Classroom discussion

VII. TYPICAL ASSIGNMENTS:

A. Homework assignments
    1. Assigned homework problems are usually from the textbook and include a mix of computational and theoretical work. Homework should take an average student 1 to 2 hours for each hour in class.
B. Problem solving laboratory. During the problem solving laboratory students work individually or in teams to solve problems related to the course material. Examples of the types of assignments given are:
    1. Quantified Predicates: Given a quantified predicate, write the negation of the statement and determine which is true, the original statement or its negation. Give a proof or counterexample, as appropriate.
    2. Digital Logic Circuits and I/O Tables: The lights in a classroom are controlled by two switches: one at the back and one at the front of the room. Moving either switch to the opposite position turns the lights off if they are on and on if they are off. Assume the lights have been installed so that when both switches are in the down position, the lights are off. Design a circuit to control the switches and construct the I/O table and Boolean expression for the circuit.
    3. Correctness of Algorithm: given an algorithm with pre- and post-conditions, use inductive methods to determine whether the algorithm is correct.
C. Writing Assignments Writing assignments can be short research papers or expository writing assignments which allow the students to delve more deeply into a particular area of interest.
    1. As an example, the instructor might provide students with a list of notable pioneers in the field of Discrete Mathematics (e.g., Lady Ada Lovelace, Augustus DeMorgan, Claude Shannon, Alan Turing) and ask each student to prepare a 2-3 page biography of one of the persons on the list. Five minutes could be set aside each class for one student to present a brief synopsis of the life of their chosen person.

VIII. EVALUATION:

Methods/Frequency

A. Exams/Tests
    Recommend two to three exams plus a comprehensive final examination
B. Quizzes
    Recommend weekly graded quizzes to monitor student progress.
C. Research Projects
    Recommend minimum of one to two writing assignments.
D. Oral Presentation
    Recommend one formal presentation
E. Group Projects
    Recommend weekly collaborative learning activities
F. Home Work
    Homework for each section covered. Recommend weekly graded homework assignments to monitor student progress.
G. Lab Activities
    Recommend weekly laboratory activities

IX. TYPICAL TEXTS:

1. Rosen, Kenneth. *Discrete Mathematics and Its Applications.* 8th ed., McGraw-Hill, 2018.
2. Johnsonbaugh, Richard. *Discrete Mathematics.* 8th ed., Pearson Prentice Hall, 2017.
3. Hunter, David. *Essentials of Discrete Mathematics* . 3rd ed., Jones & Bartlett Learning, 2015.

X. OTHER MATERIALS REQUIRED OF STUDENTS:

A. A calculator with combinatorial functions.