College of San Mateo
Official Course Outline

1. **COURSE ID:** CIS 262 **TITLE:** Discrete Mathematics for Computer Science
   Units: 3.0 units **Hours/Semester:** 48.0-54.0 Lecture hours; and 96.0-108.0 Homework hours
   Method of Grading: Grade Option (Letter Grade or Pass/No Pass)
   Prerequisite: CIS 278, or CIS 255 (or CIS 250 or CIS 284 offered at Cañada) and MATH 120, or MATH 123

2. **COURSE DESIGNATION:**
   Degree Credit
   Transfer credit: CSU; UC
   AA/AS Degree Requirements:
   CSM - COMPETENCY REQUIREMENTS: C1 Math/Quantitative Reasoning Basic Competency
   CSU GE:
   CSU GE Area B: SCIENTIFIC INQUIRY AND QUANTITATIVE REASONING: B4 - Mathematics/Quantitative Reasoning
   IGETC:
   IGETC Area 2: MATHEMATICAL CONCEPTS AND QUANTITATIVE REASONING: A: Math

3. **COURSE DESCRIPTIONS:**
   Catalog Description:
   Covers the fundamental mathematical elements of computer science including mathematical logic, sets, functions and relations, generating functions, combinatorics, methods of mathematical proof, basic number theory, algorithms, graphs, trees, basics of probability and statistics, Bayes' theorem, Binomial theorem, discrete probability, and computational complexity.

4. **STUDENT LEARNING OUTCOME(S) (SLO'S):**
   Upon successful completion of this course, a student will meet the following outcomes:
   1. Describe how formal tools of symbolic logic are used to model real-life situations, including those arising in computing contexts such as program correctness, database queries, and algorithms.
   2. Relate the ideas of mathematical induction to recursion and recursively defined structures.
   3. Demonstrate different traversal methods for trees and graphs.
   4. Use sets to solve problems in combinatorics and probability theory.

5. **SPECIFIC INSTRUCTIONAL OBJECTIVES:**
   Upon successful completion of this course, a student will be able to:
   1. Interpret mathematical notation, definitions, and theorems encountered in discrete mathematics.
   2. Write mathematical proofs using symbolic logic and Boolean Algebra.
   3. Use finite state machines to model and analyze computer operations.
   4. Formulate and interpret statements involving mathematical logic. Reformulate statements from common language to formal logic. Apply truth tables and the rules of propositional and predicate logic.
   5. Formulate short proofs applying standard techniques including direct proof, indirect proof, proof by contraposition, proof by counterexample, and mathematical induction.
   6. Demonstrate a working knowledge of set notation and elementary set theory, recognize the connection between set operations and logic, and prove elementary results involving sets.
   7. Recognize and apply the properties of functions including injective, surjective, bijective, and inverse functions.
   8. Use modular arithmetic to analyze problems involving finite mathematical structures.
   9. Solve discrete mathematics problems that involve the computation of probabilities, permutations, and combinations using fundamental enumeration principles.
   10. Apply the binomial theorem to independent events and Bayes? theorem to dependent events.

6. **COURSE CONTENT:**
   Lecture Content:
   1. Basics of logic
      A. Propositional logic
      B. The Algebra of Propositions
      C. Truth tables
      D. Normal forms

E. Equivalence and Validity
F. Predicate logic
G. Proofs and proof techniques

2. Set theory
   A. Naïve and axiomatic set theory
   B. Operations and sets
   C. Proving things about sets
   D. Countable and uncountable sets
   E. Cantor set
   F. Cantor function
3. The real numbers
   A. General properties
   B. Relations
   C. Functions
4. Basics of induction and recursion
   A. Simple induction
   B. String induction
   C. Recursive definitions
   D. Recursively defined functions
   E. Strong Induction
5. Summation notation
   A. Finite and infinite summations
   B. Finite and infinite products
   C. Convergence properties
6. Number theory
   A. Divisibility
   B. Greatest common divisor
   C. Fundamental theorem of arithmetic
   D. Modular arithmetic
   E. RSA encryption
7. Algebraic topics
   A. rings and fields
   B. vector spaces
   C. linear transformations and matrices
   D. groups
   E. semi-groups
   F. systems of equations
8. Functions
   A. Surjections
   B. Injections
   C. Inverses
   D. Composition
9. Relations
   A. Reflexivity
   B. Symmetry
   C. Transitivity
   D. Equivalence relations
10. Graphs, Trees and partial Orders
    A. Types of graphs
    B. Vertex Adjacency and Degree
    C. Some common Graphs
    D. Bipartite Graphs and Matchings
    E. Walk, Paths and connectivity
    F. Adjacency Matrices
    G. Directed Acyclic and Walk Relations
    H. Partial Orders and representing by Set Containment
    I. Linear Orders and Product Orders
    J. Equivalence Relations
    K. Forest and Trees
    L. Spanning trees/forests

M. Traversal strategies
N. K-connected Graphs

11. Basics of counting
    A. Counting arguments
    B. Sum and product rule
    C. Asymptotic
    D. Inclusion-exclusion principle
    E. Arithmetic and geometric progressions
    F. Fibonacci numbers
    G. The pigeonhole principle
12. Permutations and combinations
    A. Basic definitions
    B. The binomial theorem; binomial coefficients
    C. Pascal's identity
    D. Solving recurrence relations
    E. Common examples
    F. The Master theorem
    G. Generating functions
    H. Partial Fractions
13. Discrete probability
    A. Finite probability space; probability measure; events
    B. Conditional probability; independence; Bayes' theorem
    C. Integer random variables; expectation
    D. Independence and Distribution Functions
    E. Law of large numbers
    F. Random number generators
    G. Markov's Theorem
    H. Chebyshev's Theorem
    I. Properties of Variance
    J. Confidence in an Estimation
    K. Sums of Random Variables
14. Analysis of Algorithms and Complexity Theory
    A. rates of growth of functions,
    B. sorting and searching by pair-wise comparisons
15. Connections with linear algebra
    A. Basic matric theory
    B. Eigenvalues and eigenvectors
    C. Computational methods

7. **REPRESENTATIVE METHODS OF INSTRUCTION:**

Typical methods of instruction may include:

A. Lecture
B. Activity
C. Discussion
D. Observation and Demonstration
E. Other (Specify): Student participation in short in-class projects. Students working in small groups to solve problems.

8. **REPRESENTATIVE ASSIGNMENTS**

Representative assignments in this course may include, but are not limited to the following:

Writing Assignments:

Students will be assigned weekly homework problems from the required textbook.

Reading Assignments:

Students will read all chapters of the required textbook, reading parallel current assignments, and lecture content.

Other Outside Assignments:

Weekly homework problems
Internet research

9. **REPRESENTATIVE METHODS OF EVALUATION**

Representative methods of evaluation may include:

A. Class Participation
B. Class Performance
C. Class Work
D. Exams/Tests
E. Homework
F. Quizzes
G. Written examination

10. **REPRESENTATIVE TEXT(S):**

Possible textbooks include:

A. Eric Lehman, Thomson F. Leighton, Albert R. Meyer. *Mathematics for Computer Science*, ed. Samurai Media Limited, 2018
B. Susanna Epp. *Discrete Mathematics with Applications*, 5 ed. Cengage, 2019
C. Kenneth Rosen. *Discrete Mathematics and its Applications*, 8 ed. McGraw Hill, 2018

Origination Date: August 2020
Curriculum Committee Approval Date: November 2020
Effective Term: Fall 2021
Course Originator: Kamran Eftekhari