

# MATH 22: DISCRETE MATHEMATICS

## Foothill College Course Outline of Record

Heading	Value
<b>Effective Term:</b>	Summer 2021
<b>Units:</b>	5
<b>Hours:</b>	5 lecture per week (60 total per quarter)
<b>Prerequisite:</b>	C S 1A; MATH 48C or equivalent.
<b>Advisory:</b>	Demonstrated proficiency in English by placement via multiple measures OR through an equivalent placement process OR completion of ESLL 125 & ESLL 249; not open to students with credit in CIS 18 or C S 18.
<b>Degree &amp; Credit Status:</b>	Degree-Applicable Credit Course
<b>Foothill GE:</b>	Area V: Communication & Analytical Thinking
<b>Transferable:</b>	CSU/UC
<b>Grade Type:</b>	Letter Grade (Request for Pass/No Pass)
<b>Repeatability:</b>	Not Repeatable
<b>Cross-Listed:</b>	C S 18

## Student Learning Outcomes

- Students will apply number theory, combinatorics, discrete probability, graph theory, and recursion to solve various application problems.
- Students will develop conceptual understanding of formal logic and various methods of arguments that can be used as the basis of a computer program. They will demonstrate and communicate this understanding by writing proofs involving number theory, set theory, combinatorics, and discrete probability.
- Use formal logic and various methods of arguments to formally write proofs involving number theory, set theory, combinatorics, and discrete probability.
- Students will develop fluency in deciphering and using the language of logic, proof, and set theory, constructing logical arguments and proofs that can then be used as the basis of a computer program.

## Description

Discrete mathematics: set theory, logic, Boolean algebra, methods of proof, mathematical induction, number theory, discrete probability, combinatorics, functions, relations, recursion, algorithm efficiencies, graphs, trees.

## Course Objectives

The student will be able to:

- Use formal logic in constructing valid arguments.
- Write proofs formally, including writing proofs using symbolic logic and Boolean algebra.
- Use number theory to solve problems.
- Understand the basics of set theory, including solving problems in combinatorics and probability theory.

- Prove combination and permutation principles and use them to solve problems.
- Understand the definition of functions.
- Use recursive thinking and method to solve recurrence relations, including using recursion to analyze algorithms and programs.
- Analyze and write algorithms.
- Identify relations and their properties.
- Draw and analyze graphs and trees, including applying matrices to analyze graphs and trees.
- Solve application problems from computer science, including using finite state machines to model computer operations.
- Discuss mathematical problems and write solutions in accurate mathematical language and notation.
- Interpret mathematical solutions.

## Course Content

- Logic
  - Logical Forms and Equivalences
  - Conditional Statements
  - Valid and Invalid Arguments
  - Predicates and Quantified Statements
  - Boolean Algebra
  - Application: Digital Logic Circuits
- Methods of Proof/Proof Techniques
  - Direct Proof
  - Proof by Counterexample
  - Proof by Division into Cases
  - Proof by Contradiction and Contraposition
  - Proof by Induction
  - Strong Mathematical Induction and Well-Ordering
- Number Theory
  - Properties of Prime and Rational Numbers
  - Unique Factorization Theorem
  - Quotient-Remainder Theorem
  - Modular Arithmetic
  - Floor and Ceiling Notation
  - Applications of Number Theory to Problem Solving
  - Principle of Inclusion and Exclusion
- Set Theory
  - Notation
  - Operations on Sets
  - Cartesian Products
  - Proving Set Identities
- Counting and Probability
  - Events and Sample Space
  - Possibility Trees and Multiplication Rule
  - Addition Rule
  - Pigeonhole Principle
  - Combinations and Permutations
  - Pascal's Formula
  - Binomial Theorem
  - Discrete Probability Axioms
  - Expected Value
  - Conditional Probability, Baye's Formula
  - Integer Random Variables
  - Expectations
  - Law of Large Numbers
- Functions
  - One-to-One, Onto, Inverses
  - Compositions
  - Well Defined Functions

## G. Recursion

1. Recursively Defined Sequences
  - a. Fibonacci Numbers
2. Solving Recurrence Relations by Iteration
3. Solving Recurrence Relations using Logarithms
4. Verifying Solutions by Mathematical Induction

## H. Efficiency of Algorithms

1. Big-O, Big-Theta, and Big-Omega Notation
2. Exponential and Logarithmic Orders
3. Computing Orders of Algorithms
4. Analysis of Various Sort and Search Algorithms

## I. Relations

1. Binary Relations, N-ary Relations
2. Directed Graphs
3. Inverse Relations
4. Reflexivity, Symmetry, and Transitivity
5. Equivalence Relations and Classes

## J. Graphs and Trees

1. Definitions and Properties
2. Paths and Circuits
  - a. Euler Path
  - b. Hamiltonian Circuit
  - c. Shortest Path and Minimal Spanning Tree
3. Matrix Representation of Graphs
4. Isomorphisms of Graphs
5. Spanning Trees
6. Traversal Problems
7. Decision Trees
8. Huffman Codes
9. Warshall's Algorithm

## K. Solve Application Problems from Computer Science

1. The Application of Mathematical Induction to Recursive Computer Algorithms
2. The Use of Sequences in Loop Structures
3. The Application of Computer Logic
  - a. AND-, OR- and NOT-GATES
  - b. Boolean Algebra Structure
  - c. Logic Networks
  - d. Minimization
4. Breaking Down Problems or Functions into Components, Sub-problems or Sub-functions
5. The use of Time-complexity to Determine Big-O Growth Rate of Various Algorithms
6. Articulation Points (Cut Vertices) and Computer Networks
7. Modeling Arithmetic, Computation, and Languages, Including Algebraic Structures, Finite-state Machines and Formal Logic

## L. Discuss Mathematical Problems and Write Solutions in Accurate Mathematical Language and Notation

1. Application Problems from Other Disciplines
2. Proper Notation

## M. Interpret Mathematical Solutions

1. Explain the Significance of Solutions to Application Problems

## Lab Content

Not applicable.

## Special Facilities and/or Equipment

- A. Scientific calculator.

## Method(s) of Evaluation

- A. Written homework
- B. Quizzes, tests
- C. Proctored comprehensive final examination

## Method(s) of Instruction

- A. Lecture
- B. Discussion
- C. Cooperative learning exercises

## Representative Text(s) and Other Materials

Liben-Nowell. [Discrete Mathematics for Computer Science](#). Wiley, 2017.

Liben-Nowell. [Discrete Mathematics for Computer Science Preliminary Edition](#). Wiley, 2016.

Epp, Susanna S. [Discrete Mathematics with Applications](#). 4th ed. Brooks/Cole, 2010.

## Types and/or Examples of Required Reading, Writing, and Outside of Class Assignments

A. Homework Problems: Homework problems covering subject matter from text and related material ranging from 30-60 problems per week. Students will need to employ critical thinking in order to complete assignments.

B. Lecture: Five hours per week of lecture covering subject matter from text and related material. Reading and study of the textbook, related materials and notes.

C. Projects: Student projects covering subject matter from textbook and related materials. Projects will require students to discuss mathematical problems, write solutions in accurate mathematical language and notation and interpret mathematical solutions. Projects may require the use of a computer algebra system such as Mathematica or MATLAB.

D. Worksheets: Problems and activities covering the subject matter. Such problems and activities will require students to think critically. Such worksheets may be completed inside and/or outside of class.

## Discipline(s)

Mathematics