Public Course Outline

MATH 226 - Discrete Mathematics

Units Lecture 4.00	**Units Lab** 0.00	**Units Total** 0.00 - 4.00
Lecture Weekly Contact Hours 4.00	**Lab Weekly Contact Hours** 0.00	**Total Weekly Contact Hours** 4.00
Lecture Weekly Out of Class Hours 8.00	**Lab Weekly Outside of Class Hours** 0.00	**Total Weekly Outside of Class Hours** 8.00
Total Contact Hours 64.00 - 72.00	**Total Outside of Class Hours** 128.00 - 144.00	**Total Course Hours** 192.00 - 216.00

Typically Offered: Fall, Spring - F,SP

COURSE DESCRIPTION

Designed for students majoring in mathematics or computer science, this course introduces discrete mathematics, including logic, methods of proof, number theory, sets, counting, discrete probability, relations, recursion, recurrence relations, Boolean algebra, graphs, trees, and networks. Topics are illustrated with applications to computer science, including design and analysis of algorithms, undecidability, program correctness, and digital logic design. UC Credit Limitation: Credit for MATH 226 or MATH 226H.

ENROLLMENT RESTRICTIONS

Prerequisite
MATH 150 or MATH 150H or CS 150 or
Not open to students with prior credit in
MATH 226H

OUTLINE OF COURSE LECTURE CONTENT

The course lecture will address the following topics:
I. Propositional calculus
A. Logic of compound statements; truth tables
B. Conditional statements; contrapositive
C. Valid and invalid arguments
D. Digital logic circuits
E. Number systems and circuits for addition.

II. Predicate calculus
A. Models of a given signature
B. Quantifiers
C. Proofs
D. Divisibility of integers
E. Proofs by contradiction and contraposition
F. Euclidean algorithm.

III. Sets
A. Basic set operations; Venn diagrams; Cartesian product
B. Cardinality; pigeonhole principle
C. Russell’s paradox and halting problem

D. Boolean algebra, including applications to digital logic design
E. Minimization of circuits
F. Sequences and summation
G. Mathematical induction.

IV. Counting
A. Product rule; addition rule; principle of inclusion and exclusion
B. Permutations
C. Combinations
D. Binomial theorem
E. Finite probability space; probability measure; events
F. Conditional probability; independence; Bayes' theorem.

V. Recursion and recurrence relations
A. Recursion
B. Solution of linear, first and second order, recurrence relations
C. Use of induction to verify formulas for recursively defined sequences.
D. Use of recursion to analyze algorithms.

VI. Relations
A. Binary relations; functions
B. Reflexive, symmetric, and transitive properties
C. Equivalence relations
D. Congruences and modular arithmetic
E. Partial orders; linear orders.

VII. Graphs
A. Introduction to graphs
B. Euler paths and circuits; Hamiltonian paths; Floyd's algorithm
C. Matrix representations of graphs
D. Graph isomorphism.

VIII. Trees
A. Characterization of trees; decision trees
B. Binary trees
C. Rooted trees; Huffman code.

IX. Networks and tree algorithms
A. Spanning trees
B. Minimum spanning trees using Kruskal's algorithm
C. Minimum spanning trees using Prim's algorithm.

X. Modeling computation
A. Languages and grammars
B. Finite state machines.

PERFORMANCE OBJECTIVES

Upon successful completion of this course, students will be able to do the following:
1). Use basic logic to evaluate the correctness of mathematical arguments.
2). Apply appropriate problem-solving techniques and methods of proof to construct correct and complete proofs.
3). Solve selected problems from number theory.
4). Use Venn diagrams to solve problems about cardinalities of sets.
5). Solve counting problems using appropriate combinatorial principles.
6). Solve recurrence relations using appropriate techniques and verify the solution(s).
7). Recognize and apply linear orders, partial orders, and equivalence relations.
8). Solve selected problems about graphs.
9). Solve selected problems about trees and find a minimum spanning tree of a network.
10). Use finite state machines to model computer operations.

READING ASSIGNMENTS

Reading assignments will be consistent with, but not limited by, the following types and examples:
1). Read and solve discrete mathematics problems from the course text.
2). Read articles from books, journals, magazines, newspapers, and/or the Internet about discrete mathematics.

WRITING ASSIGNMENTS

Writing assignments will be consistent with, but not limited by, the following types and examples:
1). Write solutions to discrete mathematics homework.
2). Write formal proofs of some mathematical facts from their previous mathematical experience.
3). Write programs implementing designs from the course (recursion, graph search algorithms, minimal spanning trees).

OUTSIDE-OF-CLASS ASSIGNMENTS

Outside-of-class assignments will be consistent with, but not limited by, the following types and examples:
1). Complete reading assignments, including articles from journals, magazines, newspapers, and/or the Internet.
2). Write solutions to assigned discrete mathematics problems.
3). Write programs implementing designs from the course.

STUDENT LEARNING OUTCOMES

For a given set of problems the student will demonstrate quantitative reasoning by developing a problem-solving strategy, performing appropriate analysis and computation, and critically assessing the meaning of the conclusion or outcome.

METHODS OF INSTRUCTION

Instructional methodologies will be consistent with, but not limited by, the following types or examples:
1). Lecture by instructor, including demonstrations of solutions to discrete mathematics problems.
2). Cooperative learning through small group discussion and practice in solving discrete mathematics problems.
3). Homework assignments, including reading a discrete mathematics textbook and solving and writing out solutions to exercises and problems from the textbook.

METHODS OF EVALUATION

Evaluation methodologies will be consistent with, but not limited by, the following types or examples:
1). Performance on programming projects.
2). Quizzes and examinations that measure the student's ability to solve discrete mathematics problems and exercises using appropriate theories, principles, and techniques.
3). Homework problems, including grading correctness and completeness of solutions of computational or non-computational discrete mathematics problems and exercises.
4). Discussions and class presentations.

REQUIRED TEXTBOOKS

Examples of typical textbooks for this course include the following:

Author Epp, Susanna S

Title Discrete Mathematics with Applications

Edition 5th ed.

Publisher Cengage Learning

Year 2019

ISBN 978-1337694193

This is the most current, in-print edition. No

Author Rosen, Kenneth H
Title Discrete Mathematics and Its Applications
Edition 8th ed.
Publisher McGraw-Hill
Year 2018
ISBN 978-1259676512
This is the most current, in-print edition. No

COURSE REPEATABILITY

Total Completions Allowed: 1
Rationale for multiple enrollments:
Courses Related in Content (CRC) in Physical Education, Visual Arts, and Performing Arts:

DISTANCE ED (FORM A)

Type of Approval: 100% Online or Hybrid
You may indicate here which component(s) of the course should never be conducted online (e.g. proctored exams, labs, in-person orientation, etc.):
Requesting 100% DE in times of emergency, but during regular semesters midterm exams and the final exam should be conducted on campus or at an approved testing center.

ARTICULATION

Transfer Status: Acceptable for Credit: CSU, UC -
CSU/IGETC GE Area(s): 103 - CSU, UC

THIS COURSE IS INCORPORATED INTO THE FOLLOWING PROGRAM(S)

Computer Programming Fundamentals *ARCHIVED* Certificate of Achievement
Computer Programming Fundamentals *ARCHIVED* AA Degree
Liberal Arts with an Area of Emphasis in Mathematics and Sciences *ARCHIVED* AA Degree
Software Development *CURRENT* AA Degree
Software Development *CURRENT* Certificate of Achievement
Liberal Arts with an Area of Emphasis in Mathematics and Sciences *ARCHIVED* AA Degree
Liberal Arts with an Area of Emphasis in Mathematics and Sciences *CURRENT* AA Degree
Mathematics for Transfer *FUTURE* AS-T Degree
Software Development *FUTURE* Certificate of Achievement
Software Development *FUTURE* AA Degree