**Master Course Description for EE-321**

**Title:** Computing Fundamentals

**Credits:** 4

**Course Catalog Entry:**

**EE 321: Computing Foundations** Covers the theoretical and mathematical foundations of computation. Logic, set theory, induction, and algebraic structures with applications to computing; models of computation; limits of computability; P, NP, and NP-Complete. Prerequisite: CSE 143 and Math 126.

**Coordinator:** Sreeram Kannan, Assistant Professor, Electrical and Computer Engineering

**Goals:** To give ECE students the foundational computer science theory and discrete mathematics concepts that underpin modern computing. Provide a background in mathematical thinking and algorithm analysis. Develop a familiarity with the limits of computation, and the theoretical limits on computing performance for important classes of algorithms.

**Learning Objectives:** At the end of this course, students will be able to:

1. *Apply mathematical reasoning to computations and algorithms.*
2. *Describe algorithm growth, undecidability, and the limits of computation.*
3. *Create rigorous mathematical proofs.*

**Textbook:** Rosen, *Discrete Mathematics and Its Applications*, 6th Edition, McGraw-Hill.

**Prerequisites by Topic:**

1. Basic familiarity with computer programming (CSE 143).
2. Mature understanding of college-level mathematics (MATH 126).

**Topics:**

1. Module-1: Fundamentals (2.5 weeks)
    a. Logic and Proofs: propositional logic, equivalency, predicate logic, rules of inference, proofs & proof strategies. Rosen 1.1-1.7
    b. Set theory: sets and set operations. Rosen 2.1-2.3
2. Module-2: Algorithms and analysis (3.5 weeks)
    a. Algorithms: the growth of functions, complexity of algorithms. P, NP, NP-

    Complete Rosen 3.1-3.3

b. Induction and Recursion. Rosen 4.1-4.4

c. Counting and the Pigeonhole Principle. Rosen 5.1-5.2.

3. Module-3: Computational Models

   a. Computational Models: DFAs, NFAs, regular expressions, Turing machines.

   Rosen 12.1-12.5

b. Computability: Uncomputability, Undecidability and the Halting Problem.

c. Complexity classes and examples

**Course Structure:** The class meets for three 50-minute lectures and one 50-minute discussion section per week. The latter is administered by teaching assistants. Homework is assigned weekly. Two exams are given nominally at the ends of the 4th and 8th weeks, and a comprehensive final exam is given at the end of the quarter.

**Computer Resources:** None.

**Laboratory Resources:** None.

**Grading:** Approximate distribution: Homework 30%, Exam-1 20%, Exam-2 20%, Final Exam 30%. The grading scheme in any particular offering is the prerogative of the instructor.

**ABET Student Outcome Coverage:** This course addresses the following outcomes:

H = high relevance, M = medium relevance, L = low relevance to course.

(1) *An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics* **(M)** The homework and exams require direct application of mathematical knowledge to computer engineering problems.

(3) *An ability to communicate effectively with a range of audiences* **(L)** Students will learn and apply techniques to rigorously and formally apply and communicate theoretical concepts.

**Prepared By:** Scott Hauck, Sreeram Kannan, Linda Shapiro

**Last Revised:** May 1, 2020

# Textbook table of contents

4. Nested Quantifiers
5. Rules of Inference
6. Introduction to Proofs
7. Proof Methods and Strategy

End-of-Chapter Material

2. **Basic Structures: Sets, Functions, Sequences and Sums**
    1. Sets
    2. Set Operations
    3. Functions
    4. Sequences and Summations

End-of-Chapter Material

3. **The Fundamentals: Algorithms, the Integers, and Matrices**
    1. Algorithms
    2. The Growth of Functions
    3. Complexity of Algorithms
    4. The Integers and Division
    5. Primes and Greatest Common Divisors
    6. Integers and Algorithms
    7. Applications of Number Theory
    8. Matrices

End-of-Chapter Material

4. **Induction and Recursion**
    1. Mathematical Induction
    2. Strong Induction and Well-Ordering
    3. Recursive Definitions and Structural Induction
    4. Recursive Algorithms
    5. Program Correctness

End-of-Chapter Material

5. **Counting**
    1. The Basics of Counting
    2. The Pigeonhole Principle

3. Permutations and Combinations
4. Binomial Coefficients
5. Generalized Permutations and Combinations
6. Generating Permutations and Combinations

End-of-Chapter Material

6. **Discrete Probability**
    1. An Introduction to Discrete Probability
    2. Probability Theory
    3. Bayes' Theorem
    4. Expected Value and Variance

End-of-Chapter Material

7. **Advanced Counting Techniques**
    1. Recurrence Relations
    2. Solving Linear Recurrence Relations
    3. Divide-and-Conquer Algorithms and Recurrence Relations
    4. Generating Functions
    5. Inclusion-Exclusion
    6. Applications of Inclusion-Exclusion

End-of-Chapter Material

8. **Relations**
    1. Relations and Their Properties
    2. $n$-ary Relations and Their Applications
    3. Representing Relations
    4. Closures of Relations
    5. Equivalence Relations
    6. Partial Orderings

End-of-Chapter Material

9. **Graphs**
    1. Graphs and Graph Models
    2. Graph Terminology and Special Types of Graphs
    3. Representing Graphs and Graph Isomorphism

4. Connectivity
5. Euler and Hamilton Paths
6. Shortest-Path Problems
7. Planar Graphs
8. Graph Coloring

End-of-Chapter Material

10. **Trees**
    1. Introduction to Trees
    2. Applications of Trees
    3. Tree Traversal
    4. Spanning Trees
    5. Minimum Spanning Trees

End-of-Chapter Material

11. **Boolean Algebra**

**11.1**Boolean Functions

2. Representing Boolean Functions
3. Logic Gates
4. Minimization of Circuits

End-of-Chapter Material

12. **Modeling Computation**
    1. Languages and Grammars
    2. Finite-State Machines with Output
    3. Finite-State Machines with No Output
    4. Language Recognition
    5. Turing Machines

End-of-Chapter Material

**Appendixes**

1. Axioms for the Real Numbers and the Positive Integers
2. Exponential and Logarithmic Functions
3. Pseudocode