# *ECE 3020: Mathematics of Computer Engineering*

***Prerequisites:*** ECE 2035 (Computing Mechanisms) or ECE 2090 (Engineering Software), and Math 2401 (Calculus III) or Math 2403 (Differential Equations)

This course explores fundamental concepts in discrete mathematics and their efficient realization via appropriate algorithms, data structures, computer programs, and hardware. It also includes discussion of engineering and computational applications of the mathematical concepts.

***Course Objectives:*** As part of this course, students:

1. learn fundamental concepts of discrete mathematics
2. apply discrete math to the solution of problems in electrical and computer engineering
3. analyze algorithms and design hardware and software to realize discrete math concepts as computational artifacts

***Course Outcomes*:** Upon successful completion of this course, students should be able to:

1. use proof techniques, such as induction, to prove mathematical lemmas,
2. analyze the running times of iterative and recursive algorithms,
3. solve counting problems involving permutations, combinations, and selections,
4. apply probabilistic methods to the design and analysis of randomized algorithms,
5. design algorithms and write programs for constructing and manipulating common data abstractions, e.g. lists, trees, and graphs,
6. analyze the running times of common algorithms for trees, graphs, and networks,
7. use a context-free grammar to define the syntax of a simple programming language,
8. choose appropriate data abstractions and apply discrete math concepts in solving multiple types of electrical and computer engineering problems.

***Evaluation:*** Homeworks, programming assignments, midterm exams, final exam.

***Topical Outline:***

I. Iteration and Recursion

Iteration, mathematical induction, recursion, recurrence equations, computational complexity.

*Example applications:* parity coding, fast fourier transform, complexity analysis of recursive programs.

II. Combinatorics and Probabilistic Methods

Permutations, selections, inclusion-exclusion, probability spaces, conditional probability, independence, expectation.

*Example applications:* expected running time, Huffman codes, Monte Carlo methods, randomized algorithms.

III. Data abstractions

Trees, lists, sets, relational data, graphs.

*Example applications:* network flow, circuit partitioning and routing, decoding of error control codes.

IV. Advanced Topics

Automata theory, state minimization, regular expressions, context-free grammars.

*Example applications:* state machine design, pattern matching, parsing for compilation.