Course Name: Discrete Mathematics

Course Number: MAT* E210

Credits: 3

Catalog description: A course designed to prepare math, computer science and engineering majors for a background in abstraction, notation and critical thinking for the mathematics most directly related to computer science. Topics include: logic, relations, functions, basic set theory, proof techniques, mathematical induction, graph theory, combinatorics, discrete probability, recursion, recurrence relations, elementary number theory and graph theory.

Prerequisite: ENG*E101 and MAT*E186 (or higher) with a grade of C or higher, or appropriate placement score.

Discipline-Specific Attribute Code(s)

☒ MATH **Mathematics elective**

Course Specific Objectives:

After completing this course, the student will be able to:

1. Understand and write basic proofs (direct, contradiction, exhaustion, induction, etc.)
2. Explain and solve problems involving logic, Boolean algebra, sets, relations, functions, and recurrence relations
3. Explain and solve problems involving counting techniques including permutations, combinations, binomial theorem, and probability
4. Use logical notation to define and reason about fundamental mathematical concepts such as sets, relations, functions, and integers
5. Evaluate elementary mathematical arguments and identify fallacious reasoning as well as fallacious conclusions
6. Prove elementary properties of modular arithmetic and explain their applications in Computer Science, for example, in cryptography and hashing algorithms.
7. Demonstrate different traversal methods for trees and graphs and apply the traversal methods to application problems and model problems in Computer Science using graphs and trees

Course Content:

1. Logic and Proofs
    - Simplify and evaluate basic logic statements including compound statements, implications, inverses, converses, and contrapositives using truth tables and the properties of logic.
    - Express a logic sentence in terms of predicates, quantifiers, and logical connectives

2. Sets and Function
    - Apply the operations of sets and use Venn diagrams to solve applied problems; solve problems using the principle of inclusion-exclusion.
    - Determine the domain and range of a discrete or non-discrete function, graph functions, identify one-to-one functions, perform the composition of functions, find and/or graph the inverse of a function, and apply the properties of functions to application problems.
3. Number Theory and Cryptography
    - Use elementary number theory including the divisibility properties of numbers to determine prime numbers and composites, the greatest common divisor, and the least common multiple; perform modulo arithmetic and computer arithmetic.
4. Induction and Recursion
    - Apply rules of inference, tests for validity, and methods of proof including direct and indirect proof forms, proof by contradiction, proof by cases, and mathematical induction and write proofs using symbolic logic and Boolean Algebra.
    - Identify the base step and the recursive or inductive step in applied problems and give a recursive and a non-recursive definition for an iterative algorithm .
    - Solve problems using recurrence relations and recursion to analyze algorithms and programs such as finding Fibonacci numbers and Tower of Hanoi problems.
    - Solve problems using divide-and-conquer recurrence relations such as the fast multiplication algorithm and binary search.
5. Counting and Probability
    - Solve counting problems by applying elementary counting techniques using the product and sum rules, permutations, combinations, the pigeon-hole principle, and binomial expansion.
    - Solve discrete probability problems and use sets to solve problems in combinatorics and probability theory

6. Relations
    - Describe binary relations between two sets; determine if a binary relation is reflexive, symmetric, or transitive or is an equivalence relation; combine relations using set operations and composition.
7. Graphs and Trees
    - Determine if a given graph is simple or a multigraph, directed or undirected, cyclic or acyclic, and determine the connectivity of a graph.
    - Represent a graph using an adjacency list and an adjacency matrix and apply graph theory to application problems such as computer networks.
    - Determine if a graph has an Euler or a Hamilton path or circuit.
    - Determine if a graph is a binary tree, N-ary tree, or not a tree; use the properties of trees to classify trees, identify ancestors, descendants, parents, children, and siblings; determine the level of a node, the height of a tree or subtree and apply counting theorems to the edges and vertices of a tree.
    - Perform tree traversals using preorder, inorder, and postorder traversals and apply these traversals to application problems; use binary search trees or decision trees to solve problems.

Date Course Created: 4/26/2018

Date of Last Revision: XX/XX/20XX