COURSE SYLLABUS

Department and Course Number	CSCI 8010
Course Title	Foundations of Computer Science
Course Coordinator	Mahadevan Subramaniam msubramaniam@mail.unomaha.edu Office Hours, Tue, Thurs: 4:00 pm to 5:00 pm

1.0 Course Description

1.1 This is a foundational course for students enrolled in the graduate program in computer science. The objectives are to introduce students to a large body of concepts so that they are better prepared for undertaking the core courses in the graduate program. It is assumed that student would have programmed in a high-level language and have exposure to basic college level mathematical concepts such as logarithms, exponents, sequences, and counting principles. The course will be based on the text Foundations of Computer Science C Edition. The course will cover the chapters: 1-7 9, 12-13, 10-11.

2.0 Objectives

2.1 Students will learn a large body of concepts that provide a foundation for graduate computer science.

2.2 Students will learn about recursive algorithms, big-oh analyses, running times of programs, trees, lists, sets, and relations and graphs.

2.3 Students will learn about mathematical induction, combinatorics, discrete probability, algebra of relations, and propositional and predicate logic.

3.0 Content and Organization

3.1 Introduction
  3.1.1 Algorithms and Design of Programs
  3.1.2 C language data model
  3.1.3 Iteration, Induction and Recursion
    3.1.3.1 Iteration
    3.1.3.2 Inductive Proofs
    3.1.3.3 Complete Induction
    3.1.3.4 Proving Properties of Programs
    3.1.3.5 Recursive Functions
    3.1.3.6 Mergesort: A Recursive sorting algorithm
    3.1.3.7 Proving Properties of Recursive Programs
3.2 Running Time of Programs
  3.2.1 Choosing an Algorithm
  3.2.2 Measuring running time
  3.2.3 Big-oh and approximate running time
  3.2.4 Simplifying Big-oh expressions
  3.2.5 Analyzing Running Time
  3.2.6 Recurrence Relations and running times
3.3 Combinatorics and Probability
  3.3.1 Counting Assignments
  3.3.2 Counting Permutations

- 3.3.3 Ordered and unordered Selections
- 3.3.4 Combining Counting Rules
- 3.3.5 Probability and conditional probability
- 3.3.6 Probabilistic Reasoning
- 3.3.7 Expected Value Calculations
- 3.3.8 Programming Applications to Probability

3.4 Trees

- 3.4.1 Basic Terminology
- 3.4.2 Data structures for trees
- 3.4.3 Recursion on Trees
- 3.4.4 Binary Trees and Search Trees
- 3.4.5 Priority Queues and Partially ordered trees
- 3.4.6 Heapsort: An Application to POTs

3.5 Lists

- 3.5.1 Basic Terminology
- 3.5.2 Operations
- 3.5.3 Linked list data structure
- 3.5.4 Array-based implementation of lists
- 3.5.5 Stacks
- 3.5.6 Implementing function calls using a stack
- 3.5.7 Queues
- 3.5.8 Longest common subsequences
- 3.5.9 Representing character strings

3.6 Sets

- 3.6.1 Basic definitions
- 3.6.2 Operations
- 3.6.3 List Implementation of sets,
- 3.6.4 Characteristic-vector Implementation of sets
- 3.6.5 Hashing
- 3.6.6 Relations and Functions
- 3.6.7 Implementing functions as data
- 3.6.8 Implementing binary relations
- 3.6.9 Some special properties of binary relations

3.7 Graphs

- 3.7.1 Basic Concepts
- 3.7.2 Implementation of graphs
- 3.7.3 Connected Components
- 3.7.4 Minimal Spanning Trees
- 3.7.5 Depth-first search
- 3.7.6 Some applications of DFS

3.8 Propositional and Predicate Logics

- 3.8.1 Logical Expressions and Truth tables
- 3.8.2 Karnaugh Maps
- 3.8.3 Tautologies
- 3.8.4 Algebraic Laws for logical expressions
- 3.8.5 Deduction