EMBRY-RIDDLE AERONAUTICAL UNIVERSITY
Department of Computing and Mathematics
COURSE OUTLINE FOR

Course No.: CS222 **Title: Introduction to Discrete Structures**
Cr Hrs: 3

Lecture Hours: 3 **Laboratory Hours: 0**

COURSE DESCRIPTION:

An introduction to the fundamental algebraic, logical, and combinatorial concepts of mathematics and logic needed in subsequent computer science courses. Prerequisite: MA140 or MA120; CS115

GOALS:

This course introduces students to the basic concepts in discrete mathematics necessary for understanding more advanced computer science topics including expert systems; analysis of algorithms; relational data bases; and formal language and automata theory.

PERFORMANCE OBJECTIVES:

1. Use the basic concepts of propositional and predicate calculus to express simple English language ideas and conduct elementary logical reasoning.
2. Use basic methods of mathematical proof including proof by induction.
3. Express recurrence relations using recursive programming techniques.
4. Solve problems to indicate an understanding of basic set theory.
5. Express mathematical relations in equivalent form as tuples, graphs, and matrices.
6. Identify reflexiveness, irreflexiveness, symmetry, antisymmetry and transitively in relations.
7. Identify partial and total order in relations.
8. Form RST relations from partitions of sets and vice versa.
9. Form the transitive closure of relations.
10. Compose relations graphically and by means of matrix multiplication.
11. Identify relations which are functions including injections, surjections, and bijection.
12. Relate mathematical relations to the relational database model.
13. Form samples, selection, permutations, and combinations from a set.

14. Explain the recurrence relations implicit in Pascal's triangle.

15. Know how to count patterns in and partitions of a set.

16. Use combinatorics to solve simple word problems about picking members of a set.

17. Construct a time complexity function for simple algorithms and use the idea of asymptotic dominance to determin the order class of the algorithms.

18. Know and be able to explain the five common order classes of algorithms.

19. Apply the concept of algorithm order to categorize common sorting techniques.

20. Represent undirected graphs as tuples, graphs, and matrices.

21. Explain the graph theoretic concepts of path, cycle, and connectivity component.

22. Find eulerian paths and hamiltonian cycles in undirected graphs.

23. Determine if two simple graphs are isomorphic.

24. Give graph theoretic definitions of a tree and a forest.

25. Explain the basic graph theoretic properties of trees.

26. List the visitation order of nodes in an inorder, preorder, and postorder traversal of a binary tree.

27. Use Kruskal's and Prim's algorithms to find a minimal spanning tree for an undirected graph.

28. Apply the greedy algorithm to find a suboptimal solution of the travelling salesman problem.

29. Construct a parse tree for an assignment statement given an appropriate context free grammar.

30. Represent directed graphs as tuples, graphs, and matrices.

31. Construct the transitive closure of a digraph using matrix multiplication and Warshall's algorithms.

32. List the classes of automata and relate them to the languages they recognize.

33. Represent DFA's as 5-tuples and directed graphs.

34. Construct the next state table of a DFA from its directed graph form, and vice versa.

35. Categorize the languages accepted by simple DFA's.

36. Explain the limits of DFAs as language recognizers.

37. Represent simple Turing machines using next-state tables.

38. Depict the moves of simple Turing machines on specified input using instantaneous description triples.

39. Categorize the computations that theoretically can be performed by Turing machines as function compilers.

40. Categorize the limits of Turing machines as language recognizers.

41. Represent a simple scanner as a DFA.

Department of Computing and Mathematics
COURSE OUTLINE FOR CS220, Continued

TEXTBOOK:

Skvarcius and Robinson, *Discrete Mathematics with Computer Science Applications*, Benjamin/Cummings, 1986.

SUGGESTED SUPPLEMENTAL MATERIALS:

a. Instuctors notes

PREREQUISITE KNOWLEDGE BY TOPIC:

1. College algebra
2. Stuctured programming in a procedural language

	TOPIC	CLASS HOURS	COURSE OBJECTIVES
1.	Logic and Ssets	7	
2.	Relations and Functions	10	
3.	Combinatorics	7	
4.	Undirected Graphs	6	
5.	Directed Graphs	4	
6.	Machines and Computation	6	
7.	Hour exams	2	

LABORATORY:

COMPUTER USAGE:

GRADING SYSTEM:

ESTIMATED CONTENT:

Skills: %
Content: %