1. **TITLE OF COURSE AND COURSE NUMBER**: Discrete Structures, CS260, Credits: 3, (Major core course)

2. **DESCRIPTION OF THE COURSE**: Topics include elementary propositional and predicate logics; elementary set theory; relations and their properties; functions; congruences and Euclidean algorithm; combinatorics; mathematical reasoning; matrices; elements of graph theory; trees and their applications; Boolean algebra. Some programming will be required.

3. **COURSE PREREQUISITES**: CS 230 with a grade of C- or better

4. **COURSE OBJECTIVES**:
   - To emphasize mathematical reasoning: Students must understand mathematical reasoning in order to read, comprehend, and construct mathematical arguments.
   - To provide the ability to use combinatorial analytic methods.
   - To teach students how to work and model with discrete structures
   - To show the applications of discrete mathematics
   - To emphasize algorithmic thinking.

5. **STUDENT LEARNING OUTCOMES**:
   Upon completion of the course, the student will:
   - Be capable of formulating models and theoretical constructs needed to make further progress in Computer Science.
   - Be able to operate proficiently with these fundamental logical, combinatorial, and algebraic representations and tools in a manner required by and with a strong emphasis on Computer Scientific directions of development. Extensive practice and rigorous problem-solving drills will characterize the level of performance and expertise students will be acquiring.
   - Through classroom participation in problem-solving sessions and discussions, homework, papers, and other assignments, this course also reinforces the following student learning outcomes in accordance with the university:
     a) Effectively express themselves in precise written and oral form, both in the conciseness of mathematical formulation, first-order predicates, and in the modern standard prose. **Measure**: exams, surveys, and homework.
     b) Demonstrate the ability to think critically and logically. Students will be capable
of developing rigorous (though fundamental) mathematical proofs (on their own). They will be capable of reasoning through several methods of inference including direct, indirect, by-contradiction, enumerative, and other methods. They will be expected to formulate problems and reason by the application of both propositional and predicate logic. **Measure:** exams, surveys, and projects.

c) **Optional:** Locate and use information in the process of solving problems. **Measure:** projects.

d) **Demonstrate** the ability to integrate knowledge and ideas in a coherent and meaningful manner. **Measure:** exams and surveys.

6. **TOPICAL OUTLINE OF THE COURSE CONTENT:**

1) Logic; propositional equivalence; predicates and quantifiers.
2) Sets and set operations
3) Functions
4) Sequences and summations
5) The Growth of functions and complexity of algorithms
6) Congruences and Euclidean algorithms
7) Matrices
8) Methods of proofs
9) Mathematical induction; recursive definitions and recursive algorithms
10) Program correctness
11) Combinatorics: The basics of counting; the pigeonhole principle; permutations and combinations; discrete probability; generating permutations and combinations.
12) Recurrence relations
13) Relations: Relations and their properties; n-ary relations and their applications; representing relations; equivalence relations; partial orderings
14) Graphs: Introduction to graphs; graph terminology; representation of graphs and graph isomorphism; connectivity; Euler and Hamilton paths; shortest path; planar graphs; graph coloring.
15) Introduction to trees and their applications.
16) Boolean Algebra: Boolean functions; sum-of-products; logic gates.

7. **GUIDELINES/SUGGESTIONS FOR TEACHING METHODS AND STUDENT LEARNING ACTIVITIES:**

A. Classroom lectures and problem-solving sessions
B. Open Lab. sessions
8. GUIDELINES/SUGGESTIONS FOR METHODS OF STUDENT ASSESSMENT (OUTCOMES):

1) Periodic examinations and final examination.
2) Homework assignments.
3) Programming assignments.

9. SUGGESTED READINGS, TEXTS, OBJECTS OF STUDY:


10. BIBLIOGRAPHY OF SUPPORTIVE TEXTS AND OTHER MATERIALS:

Introductory Graph Theory, Gary Chartrand, 1985, Dover


Discrete Mathematics in Computer Science, Donald Stanat and David McAllister, 1977, Prentice Hall.


11. PREPARER'S NAME AND DATE: Dr. John Najarian

12. ORIGINAL DEPARTMENTAL APPROVAL DATE: Spring, 1996

13. REVISERS' NAME AND DATE: Dr. John Najarian, Spring 2000 and Fall 2004

14. DEPARTMENTAL REVISION APPROVAL DATE: Fall 2004