Course: Discrete Mathematics
Course Number: MATH218
Department: Mathematics

Course Description: This course is designed to give necessary mathematical background to students in computer science programs. Topics include logic, sets, basic number theory, induction and recursion, counting, relations, and graphs. Prerequisite: C- or higher in MATH217 Precalculus; waiver by placement testing results; or departmental approval.

COURSE OUTCOMES	**SAMPLE OUTCOMES ACTIVITIES**	**SAMPLE ASSESSMENT TOOLS**
Upon successful completion of this course students should:	To achieve these outcomes students may engage in the following activities:	Student learning may be assessed by:
1. Demonstrate an understanding of logic and proofs; (QL)	• Express written statements as statements using propositional logic notation • Use propositional logic to determine truth values of statements • Examine strategies for writing mathematical proofs and for avoiding mistakes in writing proofs	• Homework • In-class problem sets • Quizzes • Exams
2. Demonstrate an understanding of set notation, matrices, and functions; (QL)	• Use set notation to represent a collection of objects • Perform set operations • Examine properties and operations of functions • Develop notation for sequences and summations • Examine notation and operations involving matrices • Describe growth rates of functions using big-O notation	• Homework • In-class problem sets • Quizzes • Exams

3. Demonstrate an understanding of necessary concepts from introductory number theory; (QL)	• Perform operations using modular arithmetic • Convert numbers between different bases • Determine whether values are prime and use the Euclidean algorithm to find the greatest common divisor • Solve congruences	• Homework • In-class problem sets • Quizzes • Exams
4. Prove mathematical statements using induction; (QL)	• Determine when mathematical induction serves as an appropriate proof technique • Perform proofs by induction by constructing appropriate basis and inductive steps • Determine when strong induction or the well-ordering property should be used in proofs • Define functions recursively and perform proofs involving recursively defined functions	• Homework • In-class problem sets • Quizzes • Exams
5. Use counting techniques for problem solving; (QL)	• Use the product, sum, inclusion-exclusion, and division rules to solve counting problems • Use the Pigeonhole Principle to solve counting problems • Determine the difference between combinations and permutations and using each in solving counting problems	• Homework • In-class problem sets • Quizzes • Exams

	• Solve applications of recurrence relations	
6. Demonstrate an understanding of the relationship between elements of sets; (QL)	• Determine when a relation is reflexive, symmetric, antisymmetric, and/or transitive • Represent relations using matrices and/or graphs. • Find equivalence classes	• Homework • In-class problem sets • Quizzes • Exams
7. Demonstrate an understanding of graphs; (QL)	• Examine definitions and terminologies of basic graph theory • Construct graphs • Classify paths and connectivity of graphs • Determine whether a graph has an Euler circuit or an Euler path • Determine whether a graph has a Hamiltonian path	• Homework • In-class problem sets • Quizzes • Exams

This course includes the following core competencies: Quantitative Literacy (QL)