

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)	<ul style="list-style-type: none">Express written statements as statements using propositional logic notationUse propositional logic to determine truth values of statementsExamine strategies for writing mathematical proofs and for avoiding mistakes in writing proofs	<ul style="list-style-type: none">HomeworkIn-class problem setsQuizzesExams
2. Demonstrate an understanding of set notation, matrices, and functions; (QL)	<ul style="list-style-type: none">Use set notation to represent a collection of objectsPerform set operationsExamine properties and operations of functionsDevelop notation for sequences and summationsExamine notation and operations involving matricesDescribe growth rates of functions using big-O notation	<ul style="list-style-type: none">HomeworkIn-class problem setsQuizzesExams

<p>3. Demonstrate an understanding of necessary concepts from introductory number theory; (QL)</p>	<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • Homework • In-class problem sets • Quizzes • Exams
<p>4. Prove mathematical statements using induction; (QL)</p>	<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • Homework • In-class problem sets • Quizzes • Exams
<p>5. Use counting techniques for problem solving; (QL)</p>	<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • Homework • In-class problem sets • Quizzes • Exams

	<ul style="list-style-type: none"> • Solve applications of recurrence relations 	
6. Demonstrate an understanding of the relationship between elements of sets; (QL)	<ul style="list-style-type: none"> • Determine when a relation is reflexive, symmetric, antisymmetric, and/or transitive • Represent relations using matrices and/or graphs. • Find equivalence classes 	<ul style="list-style-type: none"> • Homework • In-class problem sets • Quizzes • Exams
7. Demonstrate an understanding of graphs; (QL)	<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • Homework • In-class problem sets • Quizzes • Exams

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