**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<br>• Use propositional logic to determine truth values of statements<br>• Examine strategies for writing mathematical proofs and for avoiding mistakes in writing proofs | • Homework<br>• In-class problem sets<br>• Quizzes<br>• Exams |
| 2. Demonstrate an understanding of set notation, matrices, and functions; (QL) | • Use set notation to represent a collection of objects<br>• Perform set operations<br>• Examine properties and operations of functions<br>• Develop notation for sequences and summations<br>• Examine notation and operations involving matrices<br>• Describe growth rates of functions using big-O notation | • Homework<br>• In-class problem sets<br>• Quizzes<br>• Exams |

| | | |
|---|---|---|
| 3. Demonstrate an understanding of necessary concepts from introductory number theory; (QL) | • Perform operations using modular arithmetic<br>• Convert numbers between different bases<br>• Determine whether values are prime and use the Euclidean algorithm to find the greatest common divisor<br>• Solve congruences | • Homework<br>• In-class problem sets<br>• Quizzes<br>• Exams |
| 4. Prove mathematical statements using induction; (QL) | • Determine when mathematical induction serves as an appropriate proof technique<br>• Perform proofs by induction by constructing appropriate basis and inductive steps<br>• Determine when strong induction or the well-ordering property should be used in proofs<br>• Define functions recursively and perform proofs involving recursively defined functions | • Homework<br>• In-class problem sets<br>• Quizzes<br>• Exams |
| 5. Use counting techniques for problem solving; (QL) | • Use the product, sum, inclusion-exclusion, and division rules to solve counting problems<br>• Use the Pigeonhole Principle to solve counting problems<br>• Determine the difference between combinations and permutations and using each in solving counting problems | • Homework<br>• In-class problem sets<br>• Quizzes<br>• 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<br>• Represent relations using matrices and/or graphs.<br>• Find equivalence classes | • Homework<br>• In-class problem sets<br>• Quizzes<br>• Exams |
| 7. Demonstrate an understanding of graphs; (QL) | • Examine definitions and terminologies of basic graph theory<br>• Construct graphs<br>• Classify paths and connectivity of graphs<br>• Determine whether a graph has an Euler circuit or an Euler path<br>• Determine whether a graph has a Hamiltonian path | • Homework<br>• In-class problem sets<br>• Quizzes<br>• Exams |

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