Modesto Junior College
Course Outline of Record

CMPSC 219

I. OVERVIEW
The following information will appear in the 2010 - 2011 catalog

CMPSC 219 Discrete Structures for Computer Science  4 Units
Prerequisite: Satisfactory completion of CMPSC 205 and MATH 121 and.

Introduction to computational topics essential for work in Computer Science. Topics include: number bases, induction, sets, relations, functions, congruence, recursion, combinations and permutations, probability, graphs, trees, logic, Boolean algebra, and proof techniques. Computing related problems and examples are integrated throughout the course.

Field trips might be required. (A-F or P/NP - Student choice) Lecture /Lab
Transfer: (CSU, UC) General Education: (MJC-GE: D2 ) (CSU-GE: B4 ) (IGETC: 2 )

II. LEARNING CONTEXT
Given the following learning context, the student who satisfactorily completes this course should be able to achieve the goals specified in Section III, Desired Learning:

A. COURSE CONTENT

1. Required Content:

   a. Number Representation
      i. Decimal (Base Ten)
      ii. Binary (Base Two)
      iii. Hexadecimal (Base 16)
      iv. Expressing Numbers in Alternate Bases

   b. Logic and Sets
      i. Statement Forms and Logical Equivalences
      ii. Set Notation
      iii. Quantifiers
      iv. Set Operations and Identities
      v. Valid Argument

   c. Basic Proof Writing
      i. Direct Demonstration
      ii. General Demonstration
      iii. Indirect Arguments
      iv. Splitting Proofs into Possible Cases
d. Elementary Number Theory
   i. Divisors
   ii. Consequences of Well-Ordering
   iii. Euclid's Algorithm
   iv. Rational and Irrational Numbers
   v. Modular Arithmetic

e. Sequences, Summation, and Induction
   i. Sequences, Indexing, and Recursion
   ii. Sigma Notation
   iii. Induction
   iv. Summation
   v. Strong Induction
   vi. The Binomial Theorem

f. Basic Relations and Functions
   i. General Relations
   ii. Basics of Functions
   iii. Special Functions
   iv. Cardinality

g. Fundamentals of Counting
   i. The Multiplication Principle
   ii. Permutations and Combinatorics
   iii. Addition and Subtraction

h. Fundamentals of Graph Theory
   i. Introduction to Graph Theory
   ii. Matrices and Special Graphs
   iii. Isomorphisms
   iv. Invariants
   v. Connectivity and Euler Circuits

i. Trees
2. **Required Lab Content:**

   a. Applications of Number Representation
      i. Decimal (Base Ten)
      ii. Binary (Base Two)
      iii. Hexadecimal (Base 16)
      iv. Expressing Numbers in Alternate Bases

   b. Diagramming Logic and Sets
      i. Statement Forms and Logical Equivalences
      ii. Set Notation
      iii. Quantifiers
      iv. Set Operations and Identities
      v. Valid Argument

   c. Investigating Basic Proofs
      i. Direct Demonstration
      ii. General Demonstration
      iii. Indirect Arguments
      iv. Splitting Proofs into Possible Cases

   d. Applying Elementary Number Theory
      i. Divisors
      ii. Consequences of Well-Ordering
      iii. Euclid's Algorithm
      iv. Rational and Irrational Numbers
      v. Modular Arithmetic

   e. Demonstrations of Sequences, Summation, and Induction
      i. Sequences, Indexing, and Recursion
      ii. Sigma Notation
      iii. Induction
iv. Summation
v. Strong Induction
vi. The Binomial Theorem

f. Using Basic Relations and Functions
   i. General Relations
   ii. Basics of Functions
   iii. Special Functions
   iv. Cardinality

g. Applications of Counting
   i. The Multiplication Principle
   ii. Permutations and Combinatorics
   iii. Addition and Subtraction

h. Applications in Graph Theory
   i. Introduction to Graph Theory
   ii. Matrices and Special Graphs
   iii. Isomorphisms
   iv. Invariants
   v. Connectivity and Euler Circuits

i. Investigating Trees
   i. Introduction to Trees
   ii. Search and Weighted Trees

B. ENROLLMENT RESTRICTIONS

1. Prerequisites
   Satisfactory completion of CMPSC 205 and MATH 121 and.

2. Requisite Skills
   Before entering the course, the student will be able to:
   a. Math level equivalent to a first semester Precalculus class.
   b. Computer Science background equivalent to ACM CS 1 course.

C. HOURS AND UNITS
### METHODS OF INSTRUCTION (TYPICAL)
Instructors of the course might conduct the course using the following method:

1. Lecture and Discussions
2. Classroom Demonstrations
3. Technology Presentations
4. Question and Answer Sessions
5. Lab Demonstrations
6. Practical Hands-On Exercises
7. Independent Study through readings

### ASSIGNMENTS (TYPICAL)

1. **EVIDENCE OF APPROPRIATE WORKLOAD FOR COURSE UNITS**
   *Time spent on coursework in addition to hours of instruction (lecture hours)*
   
   a. (Weekly) Categorize and analyze assigned topic readings.
   
   b. (Weekly) Carefully analyze and follow detailed instructions for completion of assignments.
   
   c. (Weekly) Quizzing on discrete structures concepts and terminology.
   
   d. (Weekly) Lab activities utilizing computing technology to apply discrete structure concepts.
   
   e. (Weekly) Design algorithmic solutions for discrete structure concepts.
   
   f. (Per term) Several exams at strategic points during the term.

2. **EVIDENCE OF CRITICAL THINKING**
   *Assignments require the appropriate level of critical thinking*

   a. Basic Proof: A divisibility test says that a number is divisible by 3 if and only if the sum of its digits is divisible by 3. Certainly, 3n is divisible by 3. Show for all integers 4 <= n <= 33, the tens digit and the ones digit of 3n sum to a multiple of 3.

   b. Basic Proof: Show for all m, n as elements of the integers and not both zero, the gcd(m, n) = gcd(n, m).

   c. Lab Project: Write a program to implement the Euclidean algorithm to determine the gcd of positive integers. Let a and b be positive integers. Extend the program to determine the integers s and t such that gcd(a, b) = sa + sb.

   d. Lab Project: Use Mathematica to perform the given tasks.
   i. Experiment with == for testing list equality. Specifically, what happens when \(\{1, 2, \{2, 1\}\}\) and \(\{1, 2, \{1, 2\}\}\) are compared?
   
   ii. Experiment with MemberQ for testing list membership. Specifically, determine whether \(\{\}\) is
a member of \( \{ \} \)?

e. Example Quiz/Exam Questions
i. Express the given sets in set notation: The set of integers less than -10. The set of real numbers \( x \) such that \( 3 \leq x < 7 \).
ii. Show that if \( A \) is a subset of \( B \), then \( A \) intersect \( B \) is \( A \).
iii. Compute \( 11 \) to the 10 power mod 9.
iv. Find the adjacency matrix for a graph (need actual graph) using the ordering 2, 5, 1, 4, 6, 3 of its vertices.

F. TEXTS AND OTHER READINGS (TYPICAL)


III. DESIRED LEARNING

A. COURSE GOAL

As a result of satisfactory completion of this course, the student should be prepared to:

solve problems involving sets, relations, functions, congruencies, recursively defined functions and structures, methods of combinatorics, and apply induction and other techniques to computational problems.

B. STUDENT LEARNING GOALS

Mastery of the following learning goals will enable the student to achieve the overall course goal.

1. **Required Learning Goals**

   Upon satisfactory completion of this course, the student will be able to:
   a. Perform binary and hexadecimal number conversions.
   b. Analyze mathematical induction and other techniques to prove mathematical results.
   c. Construct solutions to problems involving sets, relations, functions, and congruencies.
   d. Construct algorithmic computations using recursively defined functions and structures.
   e. Develop solutions to counting problems through methods of combinatorics.
   f. Examine the basic terminology and properties of graphs and trees.
   g. Construct solutions algorithmically for problems related to graphs and trees.
   h. Examine the logical validity of arguments and proofs as they apply to Boolean expressions.

2. **Lab Learning Goals**

   Upon satisfactory completion of the lab portion of this course, the student will be able to:
   a. Perform numeric conversions between binary, decimal, and hexadecimal numbering systems using appropriate technology.
   b. Apply mathematical induction and other techniques to prove mathematical results and explore
such techniques using technology.

c. Solve problems involving sets, relations, functions, and congruencies utilizing software tools.

d. Perform computations using recursively defined functions and structures using software tools such as recursive-compliant programming languages.

e. Use methods of combinatorics to solve counting problems through appropriate software tools.

f. Illustrate the basic terminology and properties of graphs and trees through the use of appropriate computer technology.

g. Use graphs and trees to solve problems algorithmically utilizing programming tools.

h. Investigate the logical validity of arguments and proofs as they apply to Boolean expressions and confirm such expressions in a computing environment.

IV. METHODS OF ASSESSMENT (TYPICAL)

A. FORMATIVE ASSESSMENT

1. Assignments
2. Quizzes
3. Lab Activities
4. Exams

B. SUMMATIVE ASSESSMENT

1. Assignments
2. Quizzes
3. Lab Activities
4. Exams