CATALOG INFORMATION

Dept and Nbr: MATH 4  
Title: DISCRETE MATHEMATICS  
Full Title: Discrete Mathematics  
Last Reviewed: 9/14/2020

Catalog Description:
Introductory discrete mathematics course including formal logic, Boolean logic and logic circuits, mathematical induction, introduction to number theory, set theory, principles of combinatorics, functions, relations, recursion, algorithm efficiency and graph theory.

Prerequisites/Corequisites:
Completion of MATH 27 or higher (MATH); OR Course Completion of MATH 25 and MATH 58; OR AB705 placement into Math Tier 5

Recommended Preparation:
Course Completion of MATH 1A

Limits on Enrollment:

Schedule of Classes Information:
Description: Introductory discrete mathematics course including formal logic, Boolean logic and logic circuits, mathematical induction, introduction to number theory, set theory, principles of combinatorics, functions, relations, recursion, algorithm efficiency and graph theory. (Grade Only)
Prerequisites/Corequisites: Completion of MATH 27 or higher (MATH); OR Course Completion of MATH 25 and MATH 58; OR AB705 placement into Math Tier 5
Recommended: Course Completion of MATH 1A
Limits on Enrollment:
Transfer Credit: CSU; UC.
Repeatability: Two Repeats if Grade was D, F, NC, or NP

ARTICULATION, MAJOR, and CERTIFICATION INFORMATION:

<table>
<thead>
<tr>
<th>AS Degree:</th>
<th>Area</th>
<th>Communication and Analytical Thinking</th>
<th>Effective: Fall 2001</th>
<th>Inactive:</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td></td>
<td>Math Competency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSU GE:</td>
<td>Transfer Area</td>
<td>Math/Quantitative Reasoning</td>
<td>Effective: Fall 2001</td>
<td>Inactive:</td>
</tr>
<tr>
<td>B4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGETC:</td>
<td>Transfer Area</td>
<td>Mathematical Concepts &amp; Quantitative Reasoning</td>
<td>Effective: Fall 2001</td>
<td>Inactive:</td>
</tr>
<tr>
<td>2A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSU Transfer:</td>
<td>Transferable</td>
<td>Effective: Fall 2001</td>
<td>Inactive:</td>
<td></td>
</tr>
<tr>
<td>UC Transfer:</td>
<td>Transferable</td>
<td>Effective: Fall 2001</td>
<td>Inactive:</td>
<td></td>
</tr>
</tbody>
</table>

CID:

Certificate/Major Applicable: Major Applicable Course

COURSE CONTENT

Student Learning Outcomes:
Upon completion of the course, students will be able to:
1. Recognize valid forms of arguments using predicate logic.
2. Construct mathematical proofs of propositions from elementary number theory.
3. Apply combinatorics and set theory to counting problems.
4. Analyze formal languages using finite-state automata.

Objectives:
Students will be able to:
1. Properly structure mathematical algorithms and proofs.
2. Prove theorems by induction.
3. Apply algorithms from elementary number theory.
4. Use set theory and Boolean algebra to construct and write proofs and solve problems.
5. Apply combinatorics to counting problems, including use of Pigeonhole Principle, permutations, combinations, and probability.
6. Analyze functions, inverse functions, and finite-state automata.
7. Solve recurrence relations and use recursion to analyze algorithms and programs.
8. Analyze the efficiency of algorithms.
9. Recognize relations and their properties.
10. Use graph theory and matrix representations to develop appropriate models.
11. Apply matrices to analyze graphs and trees.

**Topics and Scope:**

I. Logic
   A. Logical form, tautology, and symbolic representation in propositional logic
   B. Equivalence and minimization of Boolean circuits
   C. Valid and invalid arguments
   D. Quantified statements and predicate logic
   E. Proof strategies with number theory
   F. Logic programming

II. Mathematical Induction
   A. Sequences
   B. Weak and strong induction
   C. Well-ordering principle
   D. Correctness of algorithms

III. Combinatorics
   A. Counting
   B. Probability
   C. Possibility trees
   D. Multiplication rule
   E. Addition rule
   F. Inclusion/exclusion
   G. Permutations
   H. Combinations and Binomial Theorem
   I. Counting of multisets
   J. Pigeonhole Principle

IV. Set Theory
   A. Definitions
   B. Binary operations
   C. Properties
   D. Partitions
   E. Power sets

V. Functions
   A. Definition
   B. One-to-one, onto, and inverse functions
   C. Composition of functions

VI. Recursion
   A. Sequences defined recursively
   B. Solving recurrence relations by iteration
   C. Solutions of second-order linear homogeneous recurrence relations with constant coefficients

VII. Algorithm Efficiency
   A. Comparison of real valued functions and their graphs
   B. Big O notation
   C. Calculations of efficiency

VIII. Relations
   A. Relations on sets
   B. Reflexivity
   C. Symmetry
   D. Transitivity
E. Equivalence relations with number theory and modular arithmetic
F. Relational databases
G. Matrix representation
IX. Graph Theory
   A. Paths, Euler and Hamiltonian paths and circuits
   B. Matrix and visual representations of graphs
   C. Representations of trees, diagrams
   D. Trees and its applications: decision trees, Huffman codes
   E. Graph algorithms, including directed graphs, binary relations, shortest path and minimal spanning tree, and Warshall's algorithm (minimal weighted paths)
   F. Tree traversal algorithms
   G. Articulation points (cut vertices) and computer networks
X. Boolean Algebra Structure
   A. Logic networks
   B. Minimization
XI. Formal Languages and Automata
   A. Languages and regular expressions
   B. Finite-state automata
   C. Modeling arithmetic, computation, and languages including algebraic structures, formal languages
Assignment:
1. Reading assignments (0-50 pages per week)
2. Homework assignments (15-30) consisting of 5-35 problems from required text(s) or supplementary materials chosen by the instructor
3. Quiz(zes) (0-8)
4. Exams (2-7)
5. Final Exam
6. Project(s) (0-2): research papers on a specific topic (5-10 pages) or presentations given as posters or short talks.Papers and presentations must be related to topics taught in the course.

Methods of Evaluation/Basis of Grade:

Writing: Assessment tools that demonstrate writing skills and/or require students to select, organize and explain ideas in writing.

<table>
<thead>
<tr>
<th>Writing</th>
<th>0 - 0%</th>
</tr>
</thead>
<tbody>
<tr>
<td>None, This is a degree applicable course but assessment tools based on writing are not included because problem solving assessments are more appropriate for this course.</td>
<td>Writing</td>
</tr>
</tbody>
</table>

Problem Solving: Assessment tools, other than exams, that demonstrate competence in computational or non-computational problem solving skills.

<table>
<thead>
<tr>
<th>Problem solving</th>
<th>5 - 20%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework assignments</td>
<td>Problem solving</td>
</tr>
</tbody>
</table>

Skill Demonstrations: All skill-based and physical demonstrations used for assessment purposes including skill performance exams.
**Exams:** All forms of formal testing, other than skill performance exams.

Quiz(zes), exams, final exam

**Other:** Includes any assessment tools that do not logically fit into the above categories.

Project(s)

**Representative Textbooks and Materials:**
- Discrete Mathematics. Irani, Sandy. zyBooks. online