CS 2240 - Discrete Structures for Computer Science

Transcript/Banner Course Title: Discrete Structures for CS

College/School: College of Aerospace, Computing, Engineering, and Design

Department: Department of Computer Sciences

CIP Code: 11.0701

Check All that Apply: Required for Major

Credits: 4

Distribution of Credits: 4 + 0

Lecture: 60

Additional Student Work Hours: 120

Schedule Type(s): Lecture

Grade Mode(s): Letter

Is this course a variable topics umbrella course? No

Is this course a specified repeatable course? No

Is this course crosslisted with other prefix(es)? No

Are there course equivalencies? No

Registration Restrictions

Prerequisite(s): CS 2050 and either (MTH 1400 or equivalent) or (MTH 1120 and (MTH 1110 or equivalent)) all with grades of "C-" or better; or permission of instructor

Banner Enforced Prerequisite(s): CS 2050 AND ((College Level Math Accuplacer Score of 50 or higher) OR (Advanced Algebra and Functions [AAF] Accuplacer Score of 260 or higher) OR ((MTH 1320 or MTH 1400 or MTH 1410) with grade of C- or higher) OR ((((MTH 1109 or MTH 1110 or MTH 1111 or MTH 1112 or MTH 1115) AND (MTH 1120)) with grades of C- or higher))

Minimum Passing Grade for Banner Enforced Prerequisite(s): C- or T

Course Description:

This course provides a solid theoretical foundation for the understanding of computer science, with emphasis on the application of formal structures and reasoning to problems in computer science. The course introduces and demonstrates application of discrete mathematics concepts commonly used in computer science and needed to solve many computational problems. Topics include formal logic systems, Boolean algebra, techniques for formal reasoning (including proof methods), set theory, graph theory, functions, relations, sequences, and recursive structures.

Required reading and other materials will be equivalent to:

For each major subject of this course there is a corresponding reading list including books and on-line material such as videos, documents, and software.

The following are examples of open educational resources generally appropriate to the course:

Doerr, A. and Levasseur, K. (2020). Applied Discrete Structures. N/A: Creative Commons, http://faculty.uml.edu/klevasseur/ads2/

Smid, M. (2019) Discrete Structures for Computer Science: Counting, Recursion, and Probability. N/A: Creative Commons, http://cglab.ca/~michiel/DiscreteStructures/DiscreteStructures.pdf

DeLaplante, K. (2014) Basic Concepts in Propositional Logic. N/A: YouTube, https://www.youtube.com/playlist?list=PL619166130C21EADA

Specific, Measurable Student Behavioral Learning Objectives:

Upon completion of this course students should be able to:

1. Apply formal representations and methods of reasoning to computational problem domains.
2. Explain and construct valid arguments and proofs.
3. Interpret and use basic discrete modeling structures in computational contexts.
4. Apply fundamental computational reasoning skills.

Detailed Outline of Course Content (Major Topics and Subtopics) or Outline of Field Experience/Internship:

The pedagogy of this course revolves around concrete demonstrations of the applications of the topics through the construction of executable software programs and models.

I. Logic

- A. Precise Formal Statements
  1. Representations with correspondence to real-world domains and computational problems
- B. Truth and Falsehood
- C. Truth Tables
- D. Equivalence & Tautologies

- E. Rules of Inference
- F. Arguments
- G. Propositional Logic
- H. Predicate Logic
- I. Boolean Logic
    1. Boolean Values
    2. Boolean Operations
        a. Negation
        b. Conjunction
        c. Disjunction
        d. Exclusive Or
        e. Equality
        f. Implication
    3. Equivalences
    4. Distributive Laws

II. Formal Reasoning Applied to Computational Domains

- A. Representation
    1. Statements
    2. Proofs
    3. Theorems
    4. Lemmas
    5. Corollaries
- B. Logical Arguments and Fallacies
- C. Recognizing Valid and Invalid Proofs
- D. Counterexamples
- E. Proving Equivalences
- F. Proof by Construction
- G. Proof by Contradiction
- H. Proof by Induction

III. Number Theory & Sequences

- A. Counting
- B. Modular Arithmetic
- C. Sequences & Strings
    1. Alphabet / Symbols
    2. Length
    3. Empty String
    4. Substring
    5. Concatenation
    6. Lexicographic Order
    7. Prefix & Proper Prefix
    8. Suffix & Proper Suffix
- D. Ordered Pairs and Tuples
- E. Integers
- F. Natural Numbers
- G. Real Numbers

IV. Sets

- A. Attributes of Sets
- B. Multisets
- C. Singleton Sets
- D. Empty Set
- E. Containment
    1. Subset
    2. Proper Subset

- F. Set Operations
    1. Union
    2. Intersection
    3. Complement
    4. Difference
    5. Cartesian Product
    6. Power Set
- G. Cardinality
- H. Finite and Infinite Sets
- I. Venn Diagrams

V. Functions and Relations

- A. Mapping
- B. Domain & Range
- C. 1:1 & onto
- D. Function Arity
- E. Function Notations
- F. Predicate (Property)
- G. Binary and k-ary Relations
- H. Properties
    1. Reflexive
    2. Symmetric
    3. Transitive
- I. Equivalence Relations
- J. Recursion & Recurrence Relations

VI. Graph Theory

- A. Graphs
- B. Nodes (Vertices)
- C. Edges
- D. Undirected Graphs
- E. Directed Graphs
- F. Node Degree
    1. In-degree
    2. Out-degree
- G. Labeled Graphs
- H. Subgraphs
- I. Traversals
- J. Paths
    1. Simple Path
    2. Cycle
    3. Simple Cycle
    4. Directed Path
- K. Connected & Strongly Connected Graphs
- L. Trees
    1. Root
    2. Branch
    3. Leaf
    4. N-ary & Binary

Evaluation of Student Performance:

Student performance is evaluated through an instructor-determined subset of the following:

1. Participatory activities
2. Assignments/Projects (programming and non-programming)
3. Quizzes

4. Exams

Note: The creation date refers to the date the course was initially created. The course revision date refers to the date when the course was last revised or modified.

Course Creation Date (if known): 11/16/2020

Special Designation Information (if applicable)