School of Computing
UNIVERSITY OF GEORGIA

Course Information Sheet

CSCI 2610

Discrete Mathematics for Computer Science

Brief Course Description
(50-words or less)

This course presents a survey of the fundamental mathematical tools used in Computer Science: sets, relations, and functions; propositional and predicate logic; proof writing strategies such as direct, contradiction and induction; summations and recurrences; elementary asymptotics and timing analysis; counting and discrete probability with applications in computer science.

Extended Course Description / Comments

CSCI majors should enroll in this course MATH/CSCI 2610 instead of CSCI 2611.

Pre-Requisites and/or Co-Requisites

MATH 1113: Pre-Calculus This course cannot be taken for credit if student has received credit for MATH/CSCI 2611

Required, Elective or Selected Elective

Required Course

Approved Textbooks
(if more than one listed, the textbook used is up to the instructor's discretion)

Kenneth H. Rosen Discrete Mathematics and its Applications 7th Edition ISBN-13: 978-0073229720

Specific Learning Outcomes (Performance Indicators)

This course presents a survey of topics in discrete mathematics most relevant to students studying computer science. At the end of the semester, all students will be able to do the following:

1. Construct basic mathematical arguments using propositional logic, including structures such as quantified statements, normal form constructions, and Boolean algebra constructions.
2. Carry out set theoretic operations to describe and compare unordered collections.
3. Relate sets to functions; Use functions to describe complexity as well as basic and recursive sequences and progressions.
4. Be able to select and use an appropriate proof strategy for a potential or known theorem.
5. Carry out operations on different integer representations, and construct valid arguments for statements related to modular arithmetic.
6. Construct valid arguments for statements related to counting and discrete probability.

ABET Learning Outcomes

A. Graduates of the program will have an ability to: Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
B. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

C. Communicate effectively in a variety of professional contexts.
D. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
E. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
F. Apply computer science theory and software development fundamentals to produce computing-based solutions.

Relationship Between Student Outcomes and Learning Outcomes

		ABET Learning Outcomes					
Specific Learning Outcomes		A	B	C	D	E	F
	1		●				
	2		●				
	3		●				
	4	●	●	●			
	5		●				
	6		●				

Major Topics Covered

1. Logic and Predicates *(Knowledge level: Usage)*

a) Propositional Logic: construct compound propositions for various logic statements and natural language propositions
b) Propositional Equivalencies (Identities): use the identity laws to show propositional equivalencies and verify their truth values
c) Truth Tables: construct truth tables for compound propositions and use them as a tool for proving equivalencies
d) Normal Forms: transform compound propositions into conjunctive normal form and disjunctive normal forms, including the canonical normal forms
e) Minimization: reduce the number propositional variables and operators of a compound proposition to an equivalent proposition
f) Predicates and Quantifiers: construct compound propositions for various logic statements involving predicates and quantified statements
g) Rules of Inferences: use the laws of inference to derive a given conclusion from a set of propositions
h) Boolean Algebra and Circuits: use Boolean algebra notation and circuit diagrams to construct/visualize compound propositions

2. Set theory (Knowledge level: Usage)
   a) Sets and Operations: define set structures using different notations and define and use operations on sets
   b) Subsets and Powersets: define subsets of a set and construct the powerset.
   c) Set Identities and Equality: use the identity laws of sets to show two sets are equal
   d) Finite Cardinality: determine the cardinality of a finite set
   e) Common Sets: recognize and apply common sets in the context of set theory

f) Cartesian Products, Tuples and Relations: construct relations on sets of tuples

3. Functions (Knowledge level: Usage)
    a) Function Types: define one-to-one, onto and one-to-one correspondence functions using propositional logic; given a function prove that it is any of the three and whether it has an inverse or not
    b) Sequences: determine that a given sequence is an arithmetic or geometric progression and identify the associated constants
    c) Recursive definitions and Recurrences: construct recursive definitions for relations and sets; define the recurrence relation of sequences with its initial condition(s)
    d) Summations: understand the recursive and iterative properties of summations
    e) Cardinality of Sets, Countability and Diagonalization given a finite set determine the cardinality; given an infinite set determine whether it is countable or uncountable
    f) Given a growth function, determine its best Big-O complexity class.

4. Proof Strategies (Knowledge level: Assessment)

    a) Be able to form a valid argument from a set of assumptions using each of the following proof strategies: direct, contraposition and contradiction.
    b) Be able to form a valid argument using each of the following proof strategies: construction and counter example
    c) Induction: prove a property holds for a discrete infinite set with a smallest value
    d) Given an incorrect proof, identify the error
    e) Be able to select an appropriate proof strategy for any given theorem

5. Number Theory (Knowledge level: Usage)
    a) Divisibility and Modular Arithmetic: given two numbers find their divisor and remainder; use properties of divisibility and modular arithmetic in a proof
    b) Congruence Relations: prove the identity laws for congruence relations
    c) Integer Representations (Base Conversions): given an integer in one base convert it to another base with special focus on base 2, 8, 10 and 16
    d) Bitwise Operations: apply logic operators on bit strings
    e) Primes and composites: identify whether a number is prime or composite; identify the prime factors of a composite number
    f) Euclidean Algorithm: trace though the Euclidean algorithm to find the GCD

6. Counting (Knowledge level: Usage)

a) Rules of Counting: use the product and sum rules to determine the cardinality of a set
b) The Pigeonhole Principle: apply the Pigeonhole principle to show that there are more than one items that share a given property
c) Permutations and Combinations: determine the number of tuples or subsets of a set
d) Binomial Coefficients and Pascal's triangle: find the coefficients for a binomial expansion

7. Probability (Knowledge level: Usage)
   a) Probability Theory (Laplace): define the sample space and the event space and apply the basic rules of probability
   b) Basic Rules of Probability: apply the sum, product, conditional and independence rules
   c) Distributions: compute the probability of events in a uniform distribution
   d) Random Variables: explain potential events using the concept of random variable (Knowledge level: Familiarity)
   e) Expected Value: compute the expected value in a given trial (Knowledge level: Familiarity)
   f) Variance (time permitting): compute the variance
   g) Bayes' Theorem (time permitting): apply the Bayes' formula

Knowledge Levels

The following is the ACM's categorization of different levels of mastery: Assessment, Usage, and Familiarity. Note that Assessment encompasses both Usage and Familiarity, and Usage encompasses Familiarity.

Familiarity: The student understands what a concept is or what it means. This level of mastery concerns a basic awareness of a concept as opposed to expecting real facility with its application. It provides an answer to the question "What do you know about this?"

Usage: The student is able to use or apply a concept in a concrete way. Using a concept may include, for example, appropriately using a specific concept in a program, using a particular proof technique, or performing a particular analysis. It provides an answer to the question "What do you know how to do?"

Assessment: The student is able to consider a concept from multiple viewpoints and/or justify the selection of a particular approach to solve a problem. This level of mastery implies more than using a concept; it involves the ability to select an appropriate approach from understood alternatives. It provides an answer to the question "Why would you do that?"

Modified

01/08/2020 by Dr. Cotterell and Dr. Yazdansepas

11/05/2019 by Dr. Barnes, Dr. Cotterell, Dr. Funk, Dr. Hollingsworth and Dr. Yazdansepas

Approved

Yes