Course Outline

2026-2027 Catalog

CSCI 22 - Discrete Structures

Catalog Description

Transfer Status: CSU/UC

Prerequisite: CSCI 20 and Intermediate Algebra or equivalent

Unit(s): 3.00

Lecture: 51.00 Contact hours/102.00 Out of class hours/153.00 Total hours/3.00 Unit(s)
Total: 51.00 Contact hours/102.00 Out of class hours/153.00 Total hours/3.00 Unit(s)

Course Description: This course is an introduction to the discrete structures used in Computer Science, with an emphasis on their applications. Topics covered include functions, relations and sets, basic logic, proof techniques, basics of counting, graphs and trees, and discrete probability. (C-ID COMP 152).

Objectives

Upon successful completion of this course, the student should be able to:

1. Describe how formal tools of symbolic logic are used to model real-life situations, including those arising in computing contexts such as program correctness, database queries, and algorithms.
2. Relate the ideas of mathematical induction to recursion and recursively defined structures.
3. Analyze a problem to create relevant recurrence equations.
4. Demonstrate different traversal methods for trees and graphs.
5. Apply the binomial theorem to independent events and Bayes' theorem to dependent events.

Course Content

Topic Titles / Suggested Time Topic

Lecture

Topics	Lec Hrs
Functions, Relations, and Sets • Functions (surjections, injections, inverses, composition) • Relations (reflexivity, symmetry, transitivity, equivalence relations) • Sets (Venn diagrams, complements, Cartesian products, power sets) • Pigeonhole principles • Cardinality and countability	7.50
Basic Logic • Propositional logic • Logical connectives • Truth tables • Normal forms (conjunctive and disjunctive) • Validity • Predicate logic • Universal and existential quantification • Modus ponens and modus tollens • Limitations of predicate logic	7.50
Proof Techniques • Notions of implication, converse, inverse, contrapositive, negation, and contradiction • The structure of mathematical proofs • Direct proofs • Proof by counterexample • Proof by contradiction • Mathematical induction • Strong induction • Recursive mathematical definitions • Well orderings	12.00
Basics of Counting • Counting arguments • Sum and product rule	9.00

Topics	Lec Hrs
- Inclusion-exclusion principle - Arithmetic and geometric progressions - Fibonacci numbers - The pigeonhole principle - Permutations and combinations - Basic definitions - Pascal's identity - The binomial theorem - Solving recurrence relations - Common examples - The Master theorem	
Graphs and Trees	9.00
- Trees - Undirected graphs - Directed graphs - Spanning trees/forests - Traversal strategies	
Discrete Probability	6.00
- Finite probability space, probability measure, events - Conditional probability, independence, Bayes' theorem - Integer random variables, expectation - Law of large numbers	
Total Hours:	**51.00**

Methods of Instruction

A. Collaborative Group Work
B. Homework: Students are required to complete two hours of outside-of-class homework for each hour of lecture
C. Lecture
D. Multimedia Presentations

Methods of Evaluation

A. Quizzes
B. Projects
C. Homework
D. Mid-term and final examinations

Examples of Assignments

Reading Assignments

1. Read the section on relations (reflexivity, symmetry, transitivity and equivalence) in your text and be prepared to explain the difference between them in class.
2. Read the section in your text on trees and be prepared to provide an example in class of a spanning tree and a spanning forest.

Writing Assignments

1. Using the second principle of mathematical induction, show that any amount of postage more than one cent can be formed using just two-cent and three-cent stamps. For the basis step, note that postage of two cents can be formed using one two-cent stamp and postage of three cents can be formed using one three-cent stamp.
2. Build a truth table for DeMorgan's Laws.

Out-of-Class Assignments

1. Write a program to read in two numbers, x and n, and then compute the sum of this geometric progression: x+x^1+x^2+x^3+...+x^n. For example: if n is 3 and x is 5, then the program computes 1+5+25+125. Print x, n, and the sum. Perform error checking, for example, if the formula does not make sense for negative exponents – if n is less than 0. Have your program print an error message if n less than 0, then go back and read in the next pair of numbers without computing the sum. Are any values of x also illegal? If so, test for them, too.
2. Let set A = {1,2,3,…,n} and set B = {1,2,…,k}. Write a program that asks the user for n and k, the number of elements in sets A and B. Afterwards, display all the elements of the Cartesian Product, A x B, and the cardinal number of A x B.

Recommended Materials of Instruction

Rosen, Kenneth H. (2018). Discrete Mathematics and Its Applications. *McGraw-Hill, 8th.* 978-1259676512.

Chartrand, Gary and Zhang, Ping. (2011). Discrete Mathematics. *Waveland Press, 1st.* 978-1577667308.
Levin, Oscar. (2019). Discrete Mathematics: An Open Introduction. *Independently published, 3rd.* 978-1792901690.

Zero Cost Textbook

Applied Discrete Structures. Author: Al Doerr, Ken Levasseur. https://discretemath.org/

Minimum Qualifications

Computer Science (Masters Required)
Mathematics (Masters Required)

Created/Revised by: Sathrum, Luke
Date: 12/01/2025