TAFTCOLLEGE

Prepared by: P. Blake
Reviewed by: M. Mayfield
Date prepared: September 2022
C & GE approved: October 14, 2022
Board approved: November 9, 2022
Text Updated: September 2022
Semester effective:

Computer Science (COSC) 2100 Discrete Structures (3 Units) CSU

Prerequisite: Successful completion of ENGR 1540 Introduction to Programming Concepts and Methodologies for Engineering with Lab with a 'C' or better

Prerequisite knowledge/skills: Before entering the course, the student should be able to:

1. Describe the basics of the architecture of a computer and its components,
2. Describe the principles of structured programming,
3. Describe, design, implement, and test structured programs using currently accepted methodology, and in particular to be able to do so for programs that control or otherwise interfaces with hardware by means of software,
4. Explain what an algorithm is and its importance in computer programming, and
5. Apply numerical techniques to analyze and solve engineering-related problems.

Hours and Units Calculations:
48 hours lecture. 96 Outside of class hours. (144 Total Student Learning Hours) 3 Units

Catalog 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.

Type of Class/Course: Transfer Degree Credit

Text: Rosen, Kenneth. *Discrete Mathematics and Its Applications*, 7th Edition, McGraw Hill. 2019.

Course Objectives:

By the end of the course a successful student will be able to:

1. Describe how formal tools of symbolic logic are used to model real-life situations, include 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 Scope and Content: Lecture

Unit I Functions, Relations and Sets
- A. Functions (surjections, injections, inverses, composition)
- B. Relations (reflexivity, symmetry, transitivity, equivalence relations)

C. Sets (Venn diagrams, complements, Cartesian products, power sets)
D. Pigeonhole principles
E. Cardinality and countability

Unit II Basic Logic
A. Propositional logic
B. Logical connectives
C. Truth tables
D. Normal forms (conjunctive and disjunctive
E. Validity7
F. Predicate logic
G. Universal and existential quantification
H. Modus pones and modus tollens
I. Limitations of predicate logic

Unit III Proof Techniques
A. Notions of implications, converse, inverse, contrapositive, negation, and contradiction
B. The structure of mathematical proofs
C. Direct proofs
D. Proof by counterexample
E. Proof by contradiction
F. Mathematical induction
G. Strong induction
H. Recursive mathematical definitions
I. Well orderings

Unit IV Basics of Counting
A. Counting arguments
B. Sum and product rule
C. Inclusion-exclusion principle
D. Arithmetic and geometric progressions
E. Fibonacci numbers
F. The pigeonhole principle
G. Permutations and combinations
H. Basic definitions
I. Pascal's identity
J. The binomial theorem
K. Solving recurrence relations
L. Common examples
M. The Master theorem

Unit V Graphs and Trees
A. Trees
B. Undirected graphs
C. Directed graphs
D. Spanning trees/forests
E. Traversal strategies

Unit VI Discrete Probability
A. Finite probability space, probability measure, events

B. Conditional probability, independence, Bayes' theorem
C. Integer random variable, expectation
D. Law of Large numbers

Learning Activities Required Outside of Class:

The students in this class will spend a minimum of 6 hours per week outside of the regular class time doing the following:

1. Studying assigned text, handout materials and class notes
2. Reviewing and preparing for quizzes, midterm, and final exams
3. Completing individual homework assignments and projects following coding guidelines and proper documentation.

Methods of Instruction:

1. Lecture, demonstrations, and discussions
2. Individual and group work

Methods of Evaluation:

1. Quizzes
2. Exams
3. Participation
4. Assignments
5. Programming Projects (individual and group)
6. Class Presentations (individual and group)
7. Design Project and Presentation

Supplemental Data:

TOP Code:	070600: Computer Science
SAM Priority Code:	E: Non-Occupational
Distance Education:	Not Applicable
Funding Agency:	Y: Not Applicable (funds not used)
Program Status:	1: Program Applicable
Noncredit Category:	Y: Not Applicable, Credit Course
Special Class Status:	N: Course is not a special class

Basic Skills Status:	N: Course is not a basic skills course
Prior to College Level:	Y: Not applicable
Cooperative Work Experience:	N: Is not part of a cooperative work experience education program
Eligible for Credit by Exam:	E: Credit By Exam
Eligible for Pass/No Pass:	NO
Taft College General Education:	NONE
Discipline	Computer Science OR Engineering