BUTTE COLLEGE
COURSE OUTLINE

I. **CATALOG DESCRIPTION**

CSCI 22 - Discrete Structures **3 Unit(s)**

Prerequisite(s): CSCI 20
Recommended Prep: Reading Level IV; English Level IV; Math Level IV

Transfer Status: CSU
42.5 hours Lecture
25.5 hours Lab

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).

II. **OBJECTIVES**

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

A. 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.
B. Relate the ideas of mathematical induction to recursion and recursively defined structures.
C. Analyze a problem to create relevant recurrence equations.
D. Demonstrate different traversal methods for trees and graphs.

III. **COURSE CONTENT**

A. **Unit Titles/Suggested Time Schedule**

Lecture

Topics	Hours
1. Functions, Relations, and Sets	6.25

- Functions (surjections, injections, inverses, composition)
- Relations (reflexivity, symmetry, transitivity, equivalence relations)
- Sets (Venn diagrams, complements, Cartesian products, power sets)
- Pigeonhole principles
- Cardinality and countability

2. Basic Logic	6.25

- 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

3. Proof Techniques	10.00

- 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

4. Basics of Counting	7.50

- Counting arguments
- Sum and product rule
- 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

5. Graphs and Trees	7.50

- Trees
- Undirected graphs
- Directed graphs
- Spanning trees/forests
- Traversal strategies

6. Discrete Probability	5.00

- Finite probability space, probability measure, events
- Conditional probability, independence, Bayes' theorem
- Integer random variables, expectation
- Law of large numbers

Total Hours	42.50

Lab

Topics		Hours
1.	Functions, Relations, and Sets	3.75
2.	Basic Logic	3.75
3.	Proof Techniques	6.00
4.	Basics of Counting	4.50
5.	Graphs and Trees	4.50
6.	Discrete Probability	3.00
Total Hours		25.50

IV. **<u>METHODS OF INSTRUCTION</u>**

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

V. **<u>METHODS OF EVALUATION</u>**

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

VI. **<u>EXAMPLES OF ASSIGNMENTS</u>**

A. 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.

B. 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.

C. 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.

VII. **<u>RECOMMENDED MATERIALS OF INSTRUCTION</u>**

Textbooks:

A. Rosen, Kenneth H. <u>Discrete Mathematics and its Applications</u>. 7th Edition. McGraw-Hill, 2012.
B. Chartrand, Gary and Zhang, Ping. <u>Discrete Mathematics</u>. 1st Edition. Waveland Press, 2011.
C. Levin, Oscar. <u>Discrete Mathematics: An Open Introduction</u>. 2015 Edition. CreateSpace, 2015.

Created/Revised by: John Trolinger
Date: 03/07/2016