BUTTE COLLEGE
COURSE OUTLINE

I. CATALOG DESCRIPTION

CSCI 4 - Introduction to Programming Concepts and Methodologies   **3 Unit(s)**

Prerequisite(s): NONE

Recommended Prep: Reading Level IV; English Level IV; Math Level III

Transfer Status: CSU/UC

42.5 hours Lecture

25.5 hours Lab

This course introduces students to the fundamental concepts of programming. Students will learn about the software development life-cycle, algorithms, and the design, implementation, and testing of programs using an object-oriented programming language. (C-ID COMP 112).

II. OBJECTIVES

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

A. Describe the software development life-cycle.

B. Describe the principles of structured programming.

C. Describe, design, implement, and test structured programs using currently accepted methodology.

D. Explain what an algorithm is and its importance in computer programming.

III. COURSE CONTENT

A. Unit Titles/Suggested Time Schedule

Lecture

Topics	Hours
1. Software life-cycle including design, development, styles, documentation, testing and maintenance	2.50
2. Procedural versus objected oriented programming	2.50
• Survey of current programming languages	
3. Program design tools and programming environments	2.50
4. Documentation	2.50
5. Coding conventions	2.50
6. Data types, variables, expressions, sequential processing	2.50
7. Arrays	6.25
• Declaring and allocating arrays • Multiple-subscripted arrays	
8. Control structures	5.00
• Selective structures: if and switch • Repetitive structures: loops	
9. Algorithms including simple sorting and searching	5.00
10. File Input/Output (I/O)	2.50
• Files and streams • Sequential access files	
11. Error handling	1.25

12. Passing parameters by value and by reference	5.00
13. Principles of testing and designing test data	2.50
Total Hours	42.50

Lab

Topics	Hours
1. Program design tools and programming environments	1.50
2. Documentation	1.50
3. Coding conventions	1.50
4. Data types, variables, expressions, sequential processing	1.50
5. Arrays	3.75
6. Control structures	3.75
7. Algorithms including simple sorting and searching	3.00
8. File I/O	1.50
9. Error handling	0.75
10. Passing parameters by value and by reference	4.50
11. Principles of testing and designing test data	2.25
Total Hours	25.50

IV. **METHODS OF INSTRUCTION**
    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. Demonstrations
    E. Multimedia Presentations

V. **METHODS OF EVALUATION**
    A. Quizzes
    B. Homework
    C. Lab Projects
    D. Mid-term and final examinations

VI. **EXAMPLES OF ASSIGNMENTS**
    A. Reading Assignments
        1. Read the section in your text on loops. Be prepared to discuss the problem of infinite loops in class.
        2. Read the documentation for exception handling for the programming language you are using for your class project. Be prepared to present a short example in class.
    B. Writing Assignments
        1. Write out, in detail, an algorithm for searching for a specific value in an array of integers, and implement your algorithm in a short program.
        2. Using the sample program provided by the instructor, correct the formatting to follow the coding convention required in the class, and write documentation for all of the functions in the program. Be sure your documentation includes comments for the parameters and return value if applicable.
    C. Out-of-Class Assignments

1. Design and fully document a set of test data that would be appropriate for the sample program provided by the instructor. Be prepared to answer the following question in class: would the sample program pass or fail a test against your data?
2. Design and implement a short program that opens a text file and searches for and counts the number of occurrences of a given string. Make sure your program is case-insensitive and ignores punctuation.

VII. **RECOMMENDED MATERIALS OF INSTRUCTION**
   Textbooks:
   A. Savitch, Walter. Absolute Java. 6th Edition. Pearson, 2015.
   B. Deitel, Paul and Deitel, Harvey. C++ How to Program. 9th Edition. Prentice Hall, 2014.

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