BUTTE COLLEGE
COURSE OUTLINE

I. **CATALOG DESCRIPTION**

CSCI 20 - Programming and Algorithms I **3 Unit(s)**

Prerequisite(s): NONE
Recommended Prep: Reading Level IV; English Level IV; Math Level IV and CSCI 4
Transfer Status: CSU/UC
34 hours Lecture
51 hours Lab

This course is an introduction to the discipline of computer science, with a focus on the design and implementation of algorithms to solve simple problems using a high-level programming language. Topics include fundamental programming constructs, problem-solving strategies, debugging techniques, declaration models, and an overview of procedural and object-oriented programming languages. Students will learn to design, implement, test, and debug algorithms using pseudocode and a high-level programming language. (C-ID COMP 122).

II. **OBJECTIVES**

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

A. Design, implement, test, and debug computer programs using basic computation, simple Input/Output (I/O), standard conditional and iterative structures, and functions.
B. Use pseudocode and a high-level programming language to implement, test, and debug algorithms that solve simple problems.
C. Summarize the evolution of programming languages and describe how this evolution has led to the programming paradigms in use today.
D. Identify and demonstrate different forms of variable binding, visibility, scoping, and lifetime management.

III. **COURSE CONTENT**

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

Lecture

Topics	Hours
1. Basic syntax and semantics	4.00
2. Variables, types, expressions, and assignment	3.00
3. I/O	3.00
4. Conditional and iterative control structures	4.00
5. Functions and parameter passing	6.00
6. Structured decomposition	2.00
7. Problem-solving strategies and the role of algorithms in the problem-solving process	2.00
8. Implementation and debugging strategies for algorithms	2.00
9. Properties of algorithms	2.00
10. History of programming languages and survey of programming paradigms	1.00
11. Procedural and object-oriented programming languages	2.00
12. Variable binding, visibility, scope, lifetime, and type-checking	3.00

Total Hours	34.00

Lab

Topics	Hours
1. Basic syntax and semantics	6.00
2. Variables, types, expressions, and assignment	6.00
3. I/O	4.50
4. Conditional and iterative control structures	6.00
5. Functions and parameter passing	9.00
6. Structured decomposition	3.00
7. Problem-solving strategies	3.00
8. Implementation and debugging strategies for algorithms	6.00
9. Procedural and object-oriented programming languages	3.00
10. Variable binding, visibility, scope, lifetime, and type-checking	4.50
Total Hours	51.00

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 chapter in your textbook on dynamic arrays. Be prepared to explain in class the difference between dynamic and static allocation of arrays, and demonstrate the C++ code needed to declare, initialize, and destroy a dynamic array.
   2. Read the brief history of the Ada programming language presented online at adacore.com. Prepare a short summary of Ada language features, and compare/contrast with the features of C++.

B. Writing Assignments
   1. Write the pseudocode for the algorithm to bubble sort the contents of an integer array. Prepare a brief analysis of your algorithm, including a description of its efficiencies (or inefficiencies) and debugging techniques suitable for your algorithm.
   2. Write a short outline to decompose the following programming problem: create a program to model a game of Hangman on the computer. Your decomposition should include all user inputs, program outputs, conditional and iterative constructs, and processing required to create this program.

C. Out-of-Class Assignments
   1. Use the Internet to locate four programming languages (not including the language being

used in this class). Write a brief summary of the four languages, including key language features and the programming paradigm(s) that the languages support.

2. Visit the website freecode.com and search through the projects on the website. What do you find to be the 2-3 most popular programming languages? What do you find to be the 2-3 least popular languages? Be prepared to discuss your findings in class.

VII. RECOMMENDED MATERIALS OF INSTRUCTION

Textbooks:

A. Savitch, Walter. Absolute C++. 6th Edition. Pearson, 2015.
B. Savitch, Walter. Problem Solving with C++. 9th Edition. Pearson, 2014.
C. Gaddis, Tony. Starting Out with C++ from Control Structures to Objects. 8th Edition. Pearson, 2015.

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