GROSSMONT COLLEGE

COURSE OUTLINE OF RECORD

Curriculum Committee Approval: 11/30/2021

GCCCD Governing Board Approval: 12/14/2021

COMPUTER SCIENCE INFORMATION SYSTEMS 296 – INTRODUCTION TO C++ PROGRAMMING

1. Course Number Course Title Semester Units

CSIS 296 Introduction to C++ Programming 4

Semester Hours

4 units: 3 hours lecture: 48-54 hours 96-108 outside-of-class hours for lecture 3 hours lab: 48-54 hour

192-216 total hours

2. Course Prerequisites

None

Corequisite

None

Recommended Preparation

A “C” grade or higher or “Pass” in CSIS 119 or equivalent.

3. Catalog Description

This is an introductory course in C++ programming. Topics covered include basic language syntax, functions, data types, pointers, strings, structures, software tools, and an introduction to classes. This course is intended for persons with a prior background in any programming language.

4. Course Objectives

The student will:

a. Design and prepare programs in C++**.**

b. Create functions with parameters passed by value and by reference.

c. Analyze problems for use of proper data types.

d. Create abstract data storage.

e. Edit, compile, run and debug C++ programs within an integrated set of software tools.

f. Manipulate string data with C++.

g. Write object oriented programs using classes and objects.

5. Instructional Facilities

A computer classroom with at least one microcomputer workstation per student.

6. Special Materials Required of Student

Electronic storage media.

7. Course Content

a. Historical survey of C and C++.

b. Portability.

c. C++ program structure and language syntax.

d. Decisions (branching) in C++.

e. Iteration in C++.

f. C operators.

g. Data structures including standard data types and abstract data types.

h. Functions and pointers.

i. Input/Output.

j. Structures and unions.

k. Packaging of data structures.

l. Libraries.

m. Strings with pointers.

n. Source editors, compilers, and debuggers.

8. Method of Instruction

a. Lecture and in-class and online demonstration.

c. Student lab exercise sessions.

d. Team and individual projects in a lab environment.

9. Methods of Evaluating Student Performance

1. Examinations and quizzes, including a final examination.
2. Skills demonstration. An example could be: given a set of industrial automation requirements students will design a minimum PLC, or DCS that can control basic sensors mentioned in 7.a.1)

c. Projects and hands-on labs: an example would include the writing of a program to solve a problem with d scientific or business situation or an interactive game.

e. Outside class assignments. An example could be: In class presentation of the current trends in industrial automation.

10. Outside Class Assignments

a. Textbook reading assignments.

b. Problem solving exercises.

c. Prepare several computer programs written in C++ including documentation. Examples could include programming a desktop game, software productivity tool, or semi-complex computationally intensive programs.

11. Representative Texts

a. Representative Text(s):

Gaddis, Tony. *Starting Out with C++ From Control Structures Through Objects.* 10th edition. Pearson Publishing, 2020.

b. Supplementary texts and workbooks:

Pearson’s *My Programming Lab* for textbook in part A. <http://www.myprogramminglab.com/>

Addendum: Student Learning Outcomes

Upon completion of this course, our students will be able to do the following:

* 1. Design and prepare programs in C++ syntax.
  2. Compile, run and debug object-oriented C++ programs within an integrated set of software tools.