Computer Information
Systems Department

COP 1220 – Introduction to Programming in C++

Course Justification

This course is a required first programming C++ course in the following degrees: Associate of Arts in Computer Science, Associate in Science: Computer Programming and Analysis; Game Development; and Internet Services Technology. It is also required first programming course in C++ for the following College Credit Certificate tracks: Computer Programming and Web Development Specialist.

COP 1220 – Introduction to Programming in C++

Course Description

Introduction to Programming in "C++" covers the syntax and rules of the C++ Language. Students are required to code, compile, and execute programs. The topics of program design, structured modular programming arrays, report generation, and file processing are included. Recommended for Computer Science and Computer Information Systems majors. No previous computer courses are required although CGS 1060 is recommended. Laboratory fee. (3 hr. lecture; 2 hr. lab)

Course Competencies

Competency 1: The student will demonstrate an understanding of the program development process by:

a. Writing pseudocode for program development before writing the code.
b. Applying the techniques of functional decomposition to break a programming design problem into smaller pieces.
c. Incorporating adequate and meaningful comments into the source code of programming projects.

Competency 2: The student will demonstrate a mastery of basic C++ fundamental data types and operators by:

a. Using all the data types (float points, integers, long, double, boolean, characters, and strings) available in C++ for programming assignments.
b. Using descriptive and meaningful names in programming assignments.
c. Creating programs the use casting of data types.
d. Creating programs that use all existing operator (+,-,*, %,/,=) available in C++.
e. Explaining the properties of a variable such as its name, value, scope, persistence, and size.

COP 1220 – Introduction to Programming in C++

Competency 3: The student will demonstrate an understanding of a visual C++ programming environment by:

a. Creating C++ programs and projects in a visual C++ IDE.
b. Compiling C++ programs and projects in a visual C++ IDE.
c. Testing C++ programs and projects in a visual C++ IDE.
d. Debugging C++ programs and projects in a visual C++ IDE.
e. Executing C++ programs and projects in a visual C++ IDE.

Competency 4: The student will demonstrate an understanding of conditional statements by:

a. Creating programs that use if, else if, and else statements to evaluate conditions.
b. Creating a program that uses logical operators (and, not, or) in conditional statements.
c. Creating a program that uses comparison operators (==, <, >, <=, >=) in conditional statements.
d. Creating a program that uses the Switch, Case, and Break conditional structure to evaluate the conditions.
e. Creating a program that uses nested conditional statements.

Competency 5: The student will demonstrate an understanding of loops by:

a. Creating programs that use while, do-while and for loops to create repetition.
b. Analyzing existing programs with loops and determine the results.
c. Creating programs that use nested loops.

Competency 6: The student will demonstrate a mastery of functions by:

a. Creating functions that use call-by-reference and call-by-value.
b. Modifying existing programs that use functions.

COP 1220 – Introduction to Programming in C++

Competency 6: (Continued)

c. Creating programs that overloadfunctions.
d. Creating programs that include and use existing C++ Library functions.
e. Creating a program that uses functions to return values.
f. Identifying the scope of variables in functions.

Competency 7: The student will demonstrate an understanding of arrays by:

a. Explaining the form and uses of array.
b. Creating a program that uses single and multi-dimensioned arrays.
c. Evaluating existing programs that sort arrays.
d. Evaluating existing programs that search arrays.

Competency 8: The student will demonstrate an understanding of structures by:

a. Defining the uses of structures.
b. Creating a program that uses structures.

Competency 9: The student will demonstrate an understanding of the input and output functions of a program by:

a. Creating a program that reads an existing sequential file.
b. Creating a program that creates a sequential file.
c. Creating a program that modifies an existing sequential file.
d. Create a program that produces formatted printed output.
e. Modifying a program that produces formatted printed output.

Competency 10: The student will demonstrate an understanding of classes by:

a. Evaluating classes within existing programs.
b. Creating a program that uses classes.

COP 1220 – Introduction to Programming in C++

Competency 10: (Continued)

c. Explaining the scope of a class
d. Accessing members of a class.