CGS 2405 – Advanced Programming with C++

Course Justification

This course is the second C++ computer programming course in the Computer Science Associate in Arts degree program. This course is required for an Associate in Arts Computer Science degree as well as an Associate in Science Computer Programming and Analysis degree. The course is also required for the Computer Programming College Credit Certificate program.

CGS 2405 – Advanced Programming with C++

Course Description

This will expand on programming logic in general and C++ syntax in particular. Focus will be place on the understanding of object-oriented approach. Topics covered will include overloading operators, inheritance, advanced sorting techniques, advanced data manipulation and data structures. Students are required to design, code, compile, debug, and execute programs. This course will prepare the student for the study of advanced topics such as creating window application using C++. This course may be taken by those not majoring in Computer Science. Prerequisites: CGS1060, COP1220. Knowledge of high school algebra is recommended. Laboratory fee. (3 hr. lecture; 2 hr. lab, 4 credits)

Course Competencies

Competency 1: The student will demonstrate knowledge of data abstraction, and classes by:

a. Defining and accessing a member of a class type.
b. Writing code that invokes class member functions.
c. Implementing class member functions.
d. Understanding how encapsulation and information hiding are enforced by the C++ compiler.
e. Compiling and linking a multifile program.
f. Overriding constructor default values .
g. Overloading constructors.
h. Creating destructors.
i. Using a class within another class.
j. Creating and using private functions.
k. Explaining coupling and how to achieve loose coupling.
l. Explaining cohesion, and how to achieve high cohesion.
m. Explaining the advantages of polymorphism.

CGS 2405 – Advanced Programming with C++

Competency 2: The student will demonstrate knowledge of a linked array data structures by:

a. Creating a linked array data structure.
b. Sorting the components of a list into ascending or descending order.
c. Searching for a specific value in a sorted list using a linear search.
d. Searching for a specific value using a binary search.
e. Declaring and using strings types.

Competency 3: The student will demonstrate knowledge object oriented software development by:

a. Distinguishing between structured (procedural) programming and object-oriented programming.
b. Defining the characteristics of an object-oriented programming language.
c. Creating a new C++ class from an existing class by using inheritance.
d. Applying the object-oriented design methodology to solve a problem.
e. Using an object-oriented design and coding it in C++.
f. Creating a new C++ class from an existing class by using composition(containment).
g. Distinguishing between static and dynamic binding of operations to objects.

Competency 4: The student will demonstrate knowledge of pointers, dynamic data and reference types by:

a. Declaring variables of pointer types.
b. Taking the addresses of variables and accessing the variables through pointers.
c. Writing an expression that selects a member of a class, struct, or union using a pointer.
d. Creating and accessing dynamic data.
e. Destroying dynamic data.
f. Declaring, initializing, and accessing variables of reference types.
g. Using the "This" pointer.

CGS 2405 – Advanced Programming with C++

Competency 5: The student will demonstrate an understanding of friends classes by:

a. Declaring a friend function.
b. Using a friend function with data from two classes.
c. Using a forward reference.
d. Using a friend function with two or more instances of the same class.
e. Bestowing friendship on functions that are members of other classes.
f. Bestowing friendship on an entire class.

Competency 6: The student will demonstrate a mastery of function and operator overloading in classes by:

a. Explaining the benefits of overloading.
b. Explaining the rules that apply to operator overloading.
c. Overloading the math operators.
d. Overloading operators to work with a class object and a primitive object.
e. Overloading the insertion (<<) operator for output.
f. Overloading the extraction (>>) operator for input.
g. Overloading the prefix and postfix ++ and – operators.
h. Overloading the == operator.
i. Overloading the = operator.
j. Overloading the subscript and parentheses operators.

Competency 7: The student will demonstrate knowledge of the linked list data structure by:

a. Explaining the concept of a linked data structure.
b. Declaring the data types and variables needed for a dynamic linked list.
c. Printing the contents of a linked list.
d. Inserting new items into a linked list.
e. Deleting items from a linked list.

CGS 2405 – Advanced Programming with C++

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

a. Explaining the benefits of inheritance.
b. Creating a derived class.
c. Explaining the restrictions imposed when using inheritance.
d. Overriding and overloading parent class functions within a child class.
e. Using a constructor initialization list.
f. Overriding the class access specifier.
g. Using multiple inheritance.
h. Identifying the special problems posed by multiple inheritance.

Competency 9: The student will demonstrate knowledge of recursion by:

a. Identifying the base case(s) and the general case in a recursive definition.
b. Writing a recursive algorithm for a problem involving only simple variables.

Competency 10: The student will demonstrate an understanding of exception handling by:

a. Identifying the limitations of traditional error-handling methods.
b. Throwing exceptions.
c. Using try blocks.
d. Catching exceptions.
e. Using multiple throw statements and multiple catch blocks.
f. Throwing objects.
g. Using the default exception handler.
h. Unwinding the stack.
i. Handling memory allocation exceptions.