Miami Dade College

Course Competencies Template - Form 112

GENERAL INFORMATION	
Name: Lennie Alice Cooper	Phone #: 305 237 1224
Course Prefix/Number: COP2335	Course Title: Advanced Object Oriented Programming using C++
Number of Credits: 4	
Degree Type A.S.	☐ B.S. ☐ B.A.S ☒ A.A. ☒ A.S. ☐ A.A.S. ☐ C.C.C. ☐ A.T.C. ☐ V.C.C
Date Submitted/Revised: 07-28-2008	Effective Year/Term: 2009-1
☐ New Course Competency ☒ Revised Course Competency	
Course to be designated as a General Education course (part of the 36 hours of A.A. Gen. Ed. coursework): ☐ Yes ☒ No	

The above course links to the following Learning Outcomes:

☐ Communication
☒ Numbers / Data
☒ Critical thinking
☐ Information Literacy
☐ Cultural / Global Perspective
☐ Social Responsibility
☐ Ethical Issues
☒ Computer / Technology Usage
☐ Aesthetic / Creative Activities
☐ Environmental Responsibility

Course Description (limit to 50 words or less, **must correspond with course description on Form 102**):

This course presents advanced topics and applications of programming logic, C++ syntax, and the object-oriented approach to problem solving. Students will learn how to design, code, compile, debug, and execute Windows-based applications programs using the Windows API and Microsoft Foundation Classes (MFC). Students will learn how to apply overloading operators, inheritance, advanced sorting techniques, advanced data manipulation, and data structures. Students explore the design and use of the Open Database Connectivity (ODBC) specification. Prerequisites: CGS1060, COP1334. Knowledge of high school algebra is recommended. Laboratory fee. (4 credits).

Prerequisite(s): CGS1060, COP1334	Co requisite(s):

Course Competencies: (for further instruction/guidelines go to: http://www.mdc.edu/asa/curriculum.asp)

Competency 1: The student will demonstrate knowledge of Object Oriented Programming (OOP) Principles by:

1. Defining, creating, implementing and accessing a member of a class.
2. Explaining header files and implementation files and their uses.
3. Utilizing overriding and overloading constructors to improve program performance.
4. Creating destructors.
5. Creating and using classes.
6. Explaining the advantages of polymorphism and inheritance.
7. Utilizing overriding and overloading parent class functions within a child class.
8. Utilizing overriding the class access specifier.
9. Using a constructor initialization list.
10. Creating a new C++ class from an existing class by using both inheritance and composition (containment).
11. Distinguishing between static and dynamic binding of operations to objects.
12. Declaring and using a friend function with data from two different classes.
13. Using a friend function with two or more instances of the same class.

Revision Date: 09012008

Approved By Academic Dean Date: ______________ Reviewed By Director of Academic Programs Date: ______________

Competency 2: The student will demonstrate proficient use of arrays and vectors by:

1. Creating and using vectors.
2. Sorting the components of an array into ascending or descending order using various sorting techniques that employ the “divide and conquer” method.
3. Searching for a specific value in a sorted list using several search techniques.

Competency 3: The student will demonstrate knowledge of pointers, dynamic data and reference types by:

1. Declaring variables of the pointer type.
2. Using the ’address of operator’ and dereferencing operator.
3. Creating an expression that selects a member of a class or struct using a pointer.
4. Creating, accessing, and destroying dynamic data.
5. Declaring, initializing, and accessing variables of reference types.
6. Using the “this” pointer.

Competency 4: The student will demonstrate a mastery of function and operator overloading in classes by:

1. Explaining the benefits of overloading.
2. Explaining the rules that apply to operator overloading.
3. Applying overloading operators to improve programming performance including: math operators, the insertion and extraction operators, the prefix and postfix ++ operators, the ==and = operators, and the subscript and parentheses operators.

Competency 5: The student will demonstrate an understanding of exception handling by:

1. Identifying the limitations of traditional error-handling methods.
2. Throwing and Catching exceptions.
3. Implementing try blocks to handle data exceptions.
4. Implementing multiple throw statements and multiple catch blocks to handle data exceptions.
5. Using the default exception handler to handle data exceptions.
6. Implementing the stack unwinding concept in order to improve the performance of the program.
7. Implementing handling of memory allocation exceptions.
8. Declaring an exception class.
9. Adding an exception class to an existing class.

Competency 6: The student will demonstrate an understanding of Windows architecture and the Microsoft Foundation Classes by:

1. Using the Windows Application Programming Interface (API) to write programs for the Windows operating system.
2. Using the Microsoft Foundation Classes (MFC) to write programs for the Windows operating system.
3. Implementing View classes.

Competency 7: The student will demonstrate an understanding of a Software Development Kit (SDK) by:

1. Using SDK such as Visual Studio to develop Windows-based application programs.
2. Using an application wizard to create the MFC framework for a dialog-based application.
3. Using a Class Wizard to derive an application class from an existing Windows API class.
4. Using the Resource Editor to define a graphical user interface element for use by a Windows application.
5. Using interactive debugging and error handling tools to handle user exceptions and events.

Revision Date: 09012008

Approved By Academic Dean Date: ____________ Reviewed By Director of Academic Programs Date: ____________

Competency 8: The student will demonstrate an understanding of User Interface development by:

1. Using graphic object classes to create a graphical user interface (GUI) by: drawing text in a window, controlling text appearance, drawing simple graphic shapes.
2. Designing a visual interface using a graphics device interface.
3. Creating menus to improve program performance and accessibility.
4. Creating frame windows to improve user access to information.
5. Using window controls and messaging features to improve program functionality.
6. Creating dialog boxes to enhance user interaction.
7. Loading and manipulating bitmaps to improve program appearance.

Competency 9: The student will demonstrate an understanding of Database Applications by:

1. Explaining Record set and Data Access Objects (DAO).
2. Explaining ODBC fundamentals.
3. Describing MFC ODBC Classes.
4. Creating an ODBC application.

Revision Date: **09012008**

Approved By Academic Dean Date: ____________ Reviewed By Director of Academic Programs Date: ____________