

COURSE OUTLINE

Computer Science/Information Systems 137 Visual C++ - Object Oriented Programming and Advanced Topics

I. Catalog Statement

Computer Science/Information Systems 137 prepares the student for C++ programming in the workplace and other real world environments. The course focuses on the advanced object-oriented programming concepts needed for today's programs as well as other advanced concepts such as templates and generics, files and streams, and operator overloading. In addition, the course focuses on using the Visual C++ IDE (Integrated Development Environment) and presents some techniques for creating basic Windows-based programs in Visual C++. The course provides students an opportunity to work on projects involving graphics and game programming with the Ogre 3D graphics engine and on other advanced projects of the student's choosing within the scope of the course.

Total Lecture Units: 4.0

Total Laboratory Units: 0.0

Total Course Units: 4.0

Total Lecture Hours: 64.0

Total Laboratory Hours: 0.0

Total Laboratory Hours To Be Arranged: 0.0

Total Faculty Contact Hours: 64.0

Recommended Preparation: CS/IS 135, an equivalent C++ course, or one year of C++ programming in the workplace.

II. Course Entry Expectations

Prior to enrolling in the course, the student should be able to:

- analyze a programming task in order to develop and communicate efficient algorithms to implement that task;
- recognize programming problems on a function-by-function basis and develop structured/procedural code based on this approach;
- demonstrate an understanding of object-oriented programming concepts and object-oriented design;
- design, code, and debug basic object-based programs;
- program in the C++ language including use of objects, pointers, and structures.

III. Course Exit Standards

Upon successful completion of the required coursework, the student will be able to:

- demonstrate complete understanding of object oriented programming concepts and the application of those concepts to the C++ language;
- develop a practical understanding of techniques used in the areas of objected oriented programming, graphics, and creating a user interface;
- create software development projects.

IV. Course Content

Total Faculty Contact Hours = 64.0

- A. Introduction to Visual Studio (**3 hours**)
 - 1. Using the interface
 - 2. Creating a basic window
 - 3. Background and basics of using .NET and Visual C++
- B. Object Oriented Programming – A Deeper Look (**6 hours**)
 - 1. Const (constant) objects and const member functions
 - 2. Composition: objects as members of classes
 - 3. Friend functions and friend classes
 - 4. Dynamic memory management with operators new and delete
 - 5. Static class members
 - 6. Data abstraction and information hiding
- C. Operator Overloading; String and Array Objects (**6 hours**)
- D. Object Oriented Programming – Inheritance (**7 hours**)
 - 1. Base classes and derived classes
 - 2. Protected members
 - 3. Relationship between base classes and derived classes
 - 4. Constructors and destructors in derived classes
- E. Object Oriented Programming – Polymorphism (**7 hours**)
 - 1. Fundamentals and examples
 - 2. Virtual functions
- F. Templates and Generics (**7 hours**)
- G. Files and Streams (**7 hours**)
- H. Standard Template Library (**7 hours**)
- I. Using the Ogre 3D Graphics Library for Game Programming (**7 hours**)
 - 1. Basics of game programming
 - 2. Introduction to and use of the library
- J. Basics of Windows GUI Programming (**7 hours**)

V. Methods of Instruction

The following methods of instruction may be used in the course:

- lecture;
- demonstration;
- student presentations.

VI. Out of Class Assignments

The following out of class assignments may be used in the course:

- group and individual programming projects (e.g. move a sprite under program and algorithmic control within a window).

VII. Methods of Evaluation

The following methods of evaluation may be used in the course:

- quizzes;
- midterm examinations;
- final examination.

VIII. Textbook(s)

Deitel, Paul and Harvey Deitel. *C++: How to Program*. Upper Saddle River: Prentice Hall, 2013. Print.

12th Grade Textbook Reading Level. ISBN: 0-13-611726-1

Lippman, Stanley, Josee Lajoie and Barbara Moo. *C++ Primer*. 5th ed. Boston: Addison-Wesley Professional, 2012. Print.

10th Grade Textbook Reading Level. ISBN: 978-0321714114

IX. Student Learning Outcomes

Upon successful completion of the required coursework, the student will be able to:

- Apply both file stream and event scheduled operating system interfacing and the methods for designing programs that will successfully operate in one or both environments;
- describe the practical uses of object oriented programming with respect to database management, graphics, embedded applications and creating a user interface;
- apply programming skills to the types of problems commonly encountered in professional software development projects including intermediate and advanced object oriented programming problems.