

University College of the Cayman Islands

ASSOCIATE DEGREE SYLLABUS

COURSE NAME: COMPUTER PROGRAMMING II COURSE #: COM 201

COURSE DESCRIPTION

The purpose of this course is to develop the concepts and state-of-the-art techniques of object-oriented programming using the C++ programming language. Students will undertake programming projects to implement various concepts and principles.

Credits: 3

Prerequisite: CO 151

COURSE OBJECTIVES

This course is intended to enable the student to:

1. Attain knowledge of the C++ syntax, variables, constants and equations.
2. Understand C++ control structures and function types.
3. Implement sequential access files and structs.
4. Create and manipulate one- and two-dimensional arrays.
5. Use pointers in string manipulation, and in sorting and binary search routines.
6. Understand how to create, use and destroy class objects.
7. Establish access to object data members and member functions of classes.
8. Customize an application system by implementing added features and detailed description.
9. Understand the detailed features and principles pertaining to the construction of window, dialogs, menus and other user-interface programming objects.
10. Develop an application system that culminates with a stand-alone windows application,

COURSE OUTLINE

- I General Features of Programming in C++**
Fundamentals of object-oriented event-driven programming
Working the desktop programming environment
-Main Window, Form Window, Toolbox Window
-Project Window, Properties Window
Understanding, creating, and managing projects

- II Variables, Constants, Data Types and Equations**
Using variables to store constants

Variables declarations and initialization
Variable data types
Explicit type conversions
Named constants
Symbolic constants, pointers and strings
String Data Types
Arrays – default limits and re-initialization

III Control Structures

The *if* and *if/else* selection structure
Relational Operators
Logical Operators AND, OR and NOT
Characters and string comparison – *toupper*, *strcmp*, and *stricmp*
Nested selection structures
Application development using functions
Representing the while loop
Using counters and accumulators
Nested repetition structure
For statement and do-while loop
Break and continue statements

IV Built-In Functions and Programmer-Defined Functions

Mathematical Functions such as *pow* and *sqrt*
Void functions
Scope and lifetime variables
Passing variables by value and by reference
Value-returning functions
Function overloading
Recursion

V Arrays Handling Techniques

Creating and Initializing a one-dimensional array
Storing data in character and numeric arrays
Using parallel arrays
Searching a two-dimensional array
Binary search technique
Control Break Algorithms
Sorting a two-dimensional array

VI Pointers

Pointer operators
Calling functions by reference
Using the *Const* qualifer with Pointers
Pointer expressions and arithmetic
Pointers and arrays
Using Pointers for character and string processing

VII Classes

Class definition, scope and accessing class members
Classes and objects
Creating Constructor and Destructor functions
Initializing constructors, destructors
Using data members and member functions
Value-returning functions in class
Public and Private classes

VIII Friends, Inheritance, and Operator Overloading

Instances, and encapsulation
Constant objects and *Const* member functions
Using the *This* pointer, and operators *New* and *Delete*
Friends and static classes
Operator overloading
Overloading unary, binary operators
Overloading stream-insertion and stream-extraction operators
Inheritance types, construction, and destruction
Bases, and derived classes
Public, protected and private base classes

IX File Processing and String Stream I/O

Understanding of sequential and random access types
Opening, reading and writing a sequential access data files
Using a struct to design a record structure
Using *get*, *ignore* functions
Writing and reading multi-field records to a sequential access file
Working with bound controls
Structures and classes
Design techniques for anticipated errors
Getting error messages and error-status data
Trapping and simulating run-time errors
Centralizing error-handling in an application system

X Visual C++ Objects, Controls, and Properties

Fundamentals of objects and controls
Understanding and use of objects types
-Text box, Combo box, command button, check box
-Form, label, list box, option button, picture, frame
Relating objects types to controls
Defining objects and their properties
Classes, instances, encapsulation and inheritance
Public and private classes
Constructors and destructors

XI Dialog Boxes

Input dialog boxes
Single line input and multi-line text boxes
Types of dialog boxes

Design and displaying of dialog boxes
Modal and Modeless dialog boxes
Customizing dialog boxes – titles, properties, and command buttons

XII Objects and Form Design

An understanding of object variables
Declaration of object variables
Scope and lifetime of object variables
Object types- form and control
Creating Menus at design time
Using the Menu design window
Creating and Menu controls
Controlling Menus at run-time

ASSESSMENT

Course Assignments	-	5%
Test (x1)	-	15%
*Project	-	20%
Final Examination	-	60%
Total		100%

*The student will develop and implement a practical business application using an object-oriented/Event-driven programming language such as C++. A typical example of a project would include a “General Ledger system”.

REQUIRED TEXT

The following textbook(s) will be used for this course:

1. **Deitel H.M., Deitel P.J., C++ How to Program**, Prentice Hall

References

Grady M. Tim, C! Programming Principles and Practices, Mitchell Publishing Inc

Millspaugh A.C., Business Programming in C, Harcourt Brace

Parker, Richard, Easy Object Programming for Windows Using Visual C++, Prentice Hall

****Where applicable the latest edition of the textbook will be considered.**

Jul-08