CSC 153 C# PROGRAMMING

COURSE DESCRIPTION:

Prerequisites: None
Corequisites: None

This course introduces computer programming using the C# programming language with object-oriented programming principles. Emphasis is placed on event-driven programming methods, including creating and manipulating objects, classes, and using object-oriented tools such as the class debugger. Upon completion, students should be able to design, code, test, debug, and implement objects using the appropriate environment at the beginning level.
Course Hours Per Week: 2. Lab, 3. Semester Hours Credit, 3

LEARNING OUTCOMES:

Upon completion of this course, the student will be able to:

A. Read, write, execute, and debug C# applications
B. Understand variables and data types
C. Code decision and control structures (if, if/else, switch, while, do/while, for) and use primitive data types
D. Write user-defined methods
E. Write and manipulate arrays
F. Write programs using object-oriented programming techniques including classes, objects, inheritance, and polymorphism
G. Use graphical user interface (GUI) components
H. Understand C#’s Event Handling Model
I. Write code to access and manipulate databases

OUTLINE OF INSTRUCTION:

I. Introduction
   A. Hardware and Software
   B. How Computers Store Data
   C. How a Program Works
   D. Graphical User Interfaces
   E. Objects
   F. The Program Development Process
   G. Getting Started with the Visual Studio Environment

II. Getting Started with Forms and Controls
   A. Creating the GUI for Your First Visual C# Application
B. Introduction to C# Code
C. Writing Code for the Hello World Application
D. Label Control
E. Making Sense of IntelliSense
F. PictureBox Controls
G. Comments, Blank Lines, and Indentation
H. Writing the Code to Close an Application's Form
I. Dealing with Syntax Error

III. Reading Input with TextBox Controls
A. A First Look at Variables
B. Numeric Data Type and Variables
C. Performing Calculations
D. Inputting and Outputting Numeric Values
E. Formatting Numbers with the ToString Method
F. Simple Exception Handling
G. Using Named Constants
H. Declaring Variables as Fields
I. Using the Math Class
J. More GUI Details
K. Using the Debugger to Locate Logic Errors

IV. Decision Structures and the if Statement
A. The if-else Statement
B. Nested Decision Structures
C. Logical Operators
D. bool Variables and Flags
E. Comparing Strings
F. Preventing Data Conversion Exceptions with the TryParse Method
G. Input Validation
H. Radio Buttons and CheckBoxes
I. The switch Statement
J. Introduction to List Boxes

V. More About ListBoxes
A. The while Loop
B. The ++ and -- Operators
C. The for Loop
D. The do-while Loop
E. Using Files for Data Storage
F. The OpenFileDialog and SaveFileDialog Controls
G. Random Numbers
H. The Load Event

VI. Introduction to Methods
A. void Methods
B. Passing Arguments to Methods
C. Passing Arguments by Reference
D. Value-Returning Methods
E. Debugging Methods

VII. Value Types and Reference Types
A. Array Basics
B. Working with Files and Arrays
C. Passing Arrays as Arguments to Methods
D. Some Useful Array Algorithms
E. Advanced Algorithms for Sorting and Searching Arrays
F. Two-Dimensional Arrays
G. Jagged Arrays
H. The List Collection

VIII. More Processing Data
A. String and Character Processing
B. Structures
C. Enumerated Types
D. The ImageList Control

IX. Introduction to Classes
A. Properties
B. Parameterized Constructors and Overloading
C. Storing Class Type Object in Arrays and Lists
D. Finding the Classes and Their Responsibilities in a Problem
E. Creating Multiple Forms in a Project

X. Inheritance
A. Polymorphism
B. Abstract Classes

XI. Introduction to Database Management System
A. Tables, Rows, and Columns
B. Creating a Database in Visual Studio
C. The DataGridView Control
D. Connecting to an Existing Database and Using Details View Controls
E. More About Data-Bound Controls
F. Selecting Data with the SQL Select Statement