Silver Microsoft Partner

20483: Programming in C# (Visual Studio 2017)

Course Details

Course Outline

1. **Review of Visual C# Syntax**
    a. Overview of Writing Application by Using Visual C#
    b. Data Types, Operators, and Expressions
    c. Visual C# Programming Language Constructs
    d. Lab : Implementing Edit Functionality for the Students List
    e. Implementing Insert Functionality for the Students List
    f. Implementing Delete Functionality for the Students List
    g. Displaying a Student's Age
2. **Creating Methods, Handling Exceptions, and Monitoring Applications**
    a. Creating and Invoking Methods
    b. Creating Overloaded Methods and Using Optional and Output Parameters
    c. Handling Exceptions
    d. Monitoring Applications
    e. Lab : Extending the Class Enrolment Application Functionality
    f. Refactoring the Enrolment Code
    g. Validating Student Information
    h. Saving Changes to the Class List
3. **Basic types and constructs of Visual C#**
    a. Implementing Structs and Enums
    b. Organizing Data into Collections
    c. Handling Events
    d. Lab : Writing the Code for the Grades Prototype Application
    e. Adding Navigation Logic to the Grades Prototype Application
    f. Creating Data Types to Store User and Grade Information
    g. Displaying User and Grade Information
4. **Creating Classes and Implementing Type-Safe Collections**
    a. Creating Classes
    b. Defining and Implementing Interfaces
    c. Implementing Type-Safe Collections
    d. Lab : Adding Data Validation and Type-Safety to the Application

e. Implementing the Teacher, Student, and Grade Structs as Classes
f. Adding Data Validation to the Grade Class
g. Displaying Students in Name Order
h. Enabling Teachers to Modify Class and Grade Data

5. Creating a Class Hierarchy by Using Inheritance

a. Creating Class Hierarchies
b. Extending .NET Framework Classes
c. Lab : Refactoring Common Functionality into the User Class
d. Refactoring Common Functionality into the User Class
e. Implementing Password Complexity by Using an Abstract Method
f. Creating the ClassFullException Custom Exception

6. Reading and Writing Local Data

a. Reading and Writing Files
b. Serializing and Deserializing Data
c. Performing I/O by Using Streams
d. Lab : Generating the Grades Report
e. Serializing Data for the Grades Report as XML
f. Previewing the Grades Report
g. Persisting the Serialized Grade Data to a File

7. Accessing a Database

a. Creating and Using Entity Data Models
b. Querying Data by Using LINQ
c. Lab : Retrieving and Modifying Grade Data
d. Creating an Entity Data Model from The School of Fine Arts Database
e. Updating Student and Grade Data by Using the Entity Framework
f. Extending the Entity Data Model to Validate Data

8. Accessing Remote Data

a. Accessing Data Across the Web
b. Accessing Data by Using OData Connected Services
c. Lab : Retrieving and Modifying Grade Data Remotely
d. Creating a WCF Data Service for the SchoolGrades Database
e. Integrating the Data Service into the Application
f. Retrieving Student Photographs Over the Web (If Time Permits)

9. Designing the User Interface for a Graphical Application

a. Using XAML to Design a User Interface
b. Binding Controls to Data

c. Lab : Customizing Student Photographs and Styling the Application
d. Customizing the Appearance of Student Photographs
e. Styling the Logon View
f. Animating the StudentPhoto Control (If Time Permits)

10. Improving Application Performance and Responsiveness

a. Implementing Multitasking
b. Performing Operations Asynchronously
c. Synchronizing Concurrent Access to Data
d. Lab : Improving the Responsiveness and Performance of the Application
e. Ensuring That the UI Remains Responsive When Retrieving Teacher Data
f. Providing Visual Feedback During Long-Running Operations

11. Integrating with Unmanaged Code

a. Creating and Using Dynamic Objects
b. Managing the Lifetime of Objects and Controlling Unmanaged Resources
c. Lab : Upgrading the Grades Report
d. Generating the Grades Report by Using Word
e. Controlling the Lifetime of Word Objects by Implementing the Dispose Pattern

12. Creating Reusable Types and Assemblies

a. Examining Object Metadata
b. Creating and Using Custom Attributes
c. Generating Managed Code
d. Versioning, Signing, and Deploying Assemblies
e. Lab : Specifying the Data to Include in the Grades Report
f. Creating and Applying the IncludeInReport attribute
g. Updating the Report
h. Storing the Grades.Utilities Assembly Centrally (If Time Permits)

13. Encrypting and Decrypting Data

a. Implementing Symmetric Encryption
b. Implementing Asymmetric Encryption
c. Lab : Encrypting and Decrypting the Grades Report
d. Encrypting the Grades Report
e. Encrypting the Grades Report