Introduction to .NET Programming

Module 1: .NET Programming Fundamentals

Principles of Programming

- Writing a procedural program
- Transitioning to the object-oriented philosophy
- Coding using fundamental C# or VB syntax and semantics
- Reviewing, compiling, and executing a program in Visual Studio

Using Visual Studio

- Creating projects and solutions
- Editing and compiling a program
- Running a console application

Module 2: VB and C# Language Syntax

Coding object-oriented applications

- Dividing code into classes
- Adding fields, methods, and properties
- Defining code layout and limiting scope
- Instantiating objects

Working with data types and conversions

- Strings
- Dates and time
- Integers
- Real numbers
- Booleans
- Performing calculations with mathematical operators

- Converting between data types

Controlling program execution

- IF statements
- Writing complex criteria expressions
- CASE (switch) statements
- For...Next loops
- Do While...loops

Module 3: Programming Web Applications with ASP.NET

Constructing ASP.NET websites with Visual Studio

- Writing HTML pages and forms
- Maintaining consistency with Master Pages
- Designing pages with ASP.NET controls
- Styling sites with ASP.NET themes

Processing ASP.NET Web Forms

- Displaying data and gathering user input Activating
- Web Forms with events Maintaining variables with sessions
- Styling using CSS (Cascading Style Sheets)

Module 4: Accessing Relational Databases

Relational database concepts

- Employing databases for fast, efficient storage
- Selecting, inserting, updating, and deleting query syntax

Creating a SQL Server database

- Adding tables, fields, and indexes
- Defining primary and foreign key relationships

Accessing the database with Entity Framework

- Connecting to the database
- Storing user information

- Retrieving existing records
- Updating and saving user information
- Adding and deleting records

Module 5: More Object-Oriented Programming

Object-Oriented Program Layout

- Encapsulating data
- Ensuring proper object creation using constructors
- Extending classes using inheritance
- Overriding using virtual functions

Code Quality and Error Handling

- Ensuring type safety with enums
- Throwing and handling exceptions
- Try...Catch...Finally error handling
- Commenting code