

## Introduction to Visual Basic .NET Programming with Microsoft .NET

Length: 3 Days

### COURSE CONTENT

#### **MODULE 1: GETTING STARTED**

This module introduces Visual Basic .NET and explains how it fits into the .NET platform. It explains how to use the programming tools in Microsoft Visual Studio .NET and provides enough practice so that students can create their first application in Visual Basic .NET.

##### **Lessons:**

- Basic .NET Concepts
- Exploring the Development Environment
- Creating a Visual Basic .NET Project

**Lab:** Creating Your First Application

***After completing this module, students will be able to:***

- Begin a Visual Basic .NET project that is based on the Windows Application template.
- Use the standard toolbar, programming tools, and programming windows in the development environment to work with applications.
- Create a simple application in Visual Basic .NET.

#### **MODULE 2: WORKING WITH FORMS AND CONTROLS**

This module explains fundamental programming concepts, including event-driven programming, classes, objects, properties, methods, and events. This module also explains how to use forms and controls to create a user interface. This includes the following: how to create a form, set properties, call methods, and write code for events; how to add controls to a form; how to manage multiple forms; how to use message boxes; how to use naming conventions; and how to format and document code.

##### **Lessons**

- Understanding Programming Concepts
- Working with Windows Forms
- Working with Controls
- Styling Your Code

**Lab:** Creating the User Interface

***After completing this module, students will be able to:***

- Explain fundamental programming concepts, including event-driven programming, classes, objects, properties, methods, and events.
- Use forms and controls to create a user interface.
- Create and manage multiple forms in a simple application.
- Interact with a user by using the MessageBox function.
- Use Visual Basic naming conventions and standards for formatting and documenting code.

### **MODULE 3: USING VARIABLES AND ARRAYS**

This module explains how to name, declare, assign values to, and use variables and constants. It explains how to declare variables with different levels of scope, how to create your own data structures, and how to convert variable values from one data type to another. It also describes how to store data in an array.

#### **Lessons**

- Introduction to Data Types
- Using Variables
- Variable Scope
- Converting Data Types
- Creating and Using Structures
- Storing Data in Arrays

**Lab:** Creating and Using Variables

**Lab:** Using Structures and Arrays

**After completing this module, students will be able to:**

- Describe the various data types they can use to declare variables.
- Name, declare, initialize, and use variables and constants.
- Declare variables with different levels of scope.
- Create user-defined structures.
- Convert variable values from one data type to another.
- Store data in arrays

### **MODULE 4: WORKING WITH PROCEDURES**

This module describes how to create and use Sub and Function procedures, including predefined functions, and how to structure code for increased reusability.

#### **Lessons**

- Creating Procedures
- Using Procedures
- Using Predefined Functions

**Lab:** Creating and Using Procedures

**After completing this module, students will be able to:**

- Create and call Sub procedures and Function procedures.
- Write procedures in modules to enable code reuse.
- Pass arguments by value and by reference.
- Use predefined functions in application code.

### **MODULE 5: DECISION STRUCTURES AND LOOPS**

This module explains how to implement decision structures and loop structures to control program output and execution.

#### **Lessons**

- Using Conditional Expressions
- Using Decision Structures
- Using Conditional Loop Structures

**Lab:** Using Decision Structures

**After completing this module, students will be able to:**

- Create formulas and conditional expressions by using arithmetic, comparison, and logical operators.
- Use If Then structures to evaluate whether a condition is true or false and direct the program flow accordingly.
- Use Select Case structures to execute one of several statements.
- Use For Next structures to execute statements a set number of times.
- Use Do Loop structures to execute statements until a specific condition is met or while a specific condition is true.
- Choose the appropriate decision structure or loop based on the requirements of the application.

## **MODULE 6: VALIDATING USER INPUT**

This module explains how to validate user input at both the field level and the form level. It describes how to handle invalid input by providing error messages and guiding users through the process of finding and fixing errors. It describes how to use control properties and methods to restrict and validate data entry.

### **Lessons**

- Restricting User Input
- Validating Field Data
- Validating Form Data

**Lab:** Validating User Input

***After completing this module, students will be able to:***

- Restrict the type of data that can be entered in a field.
- Test user input at the field level to determine if it is valid, and display messages to help the user correct invalid data.
- Set control properties to specify the order of data entry, the type of data to enter, and how to display the data when the application is run
- Validate user input at the form level, and guide users through the process of finding and fixing errors.

## **MODULE 7: OBJECT-ORIENTED PROGRAMMING IN VISUAL BASIC .NET**

This module explains how to create and use classes. The module explains the concepts of abstraction, encapsulation, instantiation, initialization, constructors, and destructors. This module also describes inheritance, polymorphism, and namespaces.

### **Lessons**

- Understanding Classes
- Working with Classes
- Using Shared Members
- Inheritance, Polymorphism, and Namespaces

**Lab:** Creating a Derived Class

***After completing this module, students will be able to:***

- Explain object-oriented programming concepts, including abstraction, encapsulation, classes, and objects
- Use the Object Browser to examine available programming elements, including classes and objects.
- Create a new class, including its methods, properties, and data members with appropriate access levels.
- Create and use an instance of a class, including instance and shared data members, and shared and non-shared methods
- Explain how constructors and destructors work.
- Explain inheritance, polymorphism, and namespaces.

## **MODULE 8: HANDLING ERRORS AND EXCEPTIONS**

This module explains types of errors that can occur in a program and explains how to use the debugging tools provided with Visual Basic .NET to help diagnose and correct the errors. These tools include the Visual Studio .NET debugger, debugging windows, and structured exception handling.

### **Lessons**

- Types of Errors
- Using the Debugger
- Handling Exceptions

**Lab:** Implementing Structured Exception Handling

***After completing this module, students will be able to:***

- Define and give examples of syntax, run-time, and logic errors.
- Debug code by using the Visual Studio .NET debugger.
- Implement structured exception handling in an application.

## **MODULE 9: ENHANCING THE USER INTERFACE**

This module explains how to create menus, status bars, and toolbars to enhance the usability of an application.

### **Lessons**

- Creating Menus
- Creating Status Bars
- Creating Toolbars

### ***After completing this module, students will be able to:***

- Create custom menus to group application commands.
- Create a status bar to provide users with feedback about an application.
- Create a toolbar to provide a graphical interface with which users can access key functions of an application.

## **MODULE 10: WEB FORMS AND XML WEB SERVICES**

This module explains how to create a Web Forms application and how to invoke a simple XML Web service.

### **Lessons**

- Working with Web Forms
- Using XML Web Services

**Lab:** Creating a Web Application

### ***After completing this module, students will be able to:***

- Create, build, and run an application that uses Web Forms.
- Explain the process for discovering and calling XML Web services.
- Use a proxy to call an XML Web service from a Web Form.

## **MODULE 11: USING ADO.NET**

This module explains how to use ADO.NET with a Windows Forms application to create, read, update, and delete records in Access and SQL Server databases.

### **Lessons**

- Database Concepts
- Overview of ADO.NET
- Working with Data

**Lab:** Accessing Data with ADO.NET

### ***After completing this module, students will be able to:***

- Define basic database terminology, including database, table, record, field, and key.
- Describe some commonly used ADO.NET objects.
- Create and open a connection to a database.
- Create, read, update, and delete records in a database.
- Use the Data Form Wizard to create a simple data access application.
- Display and modify data extracted from a database.

## **MODULE 12: DEPLOYING APPLICATIONS**

This module explains how to deploy applications by using Visual Studio .NET. The module also describes deployment options available in Visual Basic .NET, and how to create and configure a setup project for a Windows-based application.

### **Lessons**

- Introduction to Deployment
- Deploying a Windows-based Application

**Lab:** Deploying an Application

### ***After completing this module, students will be able to:***

- Describe how to deploy applications by using Visual Studio .NET.
- Create and customize a Setup program for a Windows-based application