Introduction to Visual Basic .NET Programming with Microsoft .NET

Course 2559—Five days, Instructor-led

Introduction

Elements of this syllabus are subject to change.

This five-day instructor-led course provides students with the knowledge and skills needed to develop applications in Microsoft Visual Basic .NET for the Microsoft .NET platform. The course focuses on user interfaces, program structure, language syntax, and implementation details. This is the first course in the Visual Basic .NET curriculum and will serve as the entry point for other .NET courses.

Audience

This course is intended for both novice and experienced programmers who have a minimum of three months programming experience and have basic Microsoft Windows navigation skills.

At Course Completion

After completing this course, students will be able to:

- Create a simple Visual Basic .NET–based application based on the Windows Application template.
- Use forms and controls to create a user interface.
- Create and use variables and arrays.
- Create and use Sub and Function procedures, including predefined functions.
- Implement decision structures and loops by using conditional expressions.
- Validate user input for fields, controls, and forms.
- Apply object-oriented programming techniques to create classes, add methods, and add properties.
- Resolve syntax, run-time, and logic errors by using the debugger and structured exception handling.
- Enhance the user interface by adding menus, status bars, and toolbars.
- Create a simple Visual Basic .NET–based Web Forms application that uses an XML Web Service.
- Access and manipulate data in a Microsoft Access or Microsoft SQL Server™ database by using Microsoft ADO.NET.
- Build, package, and deploy an application.

Prerequisites

Before attending this course, students must have:

- Windows navigation skills.
- Experience (at least three months) with developing applications in either a graphical or non-graphical environment.

– And –

- Completion of Microsoft MSDN Training Course 2667: Introduction to Programming.

– Or –

- Understanding of the basics of structured programming, including concepts such as flow control, variables and parameters, and function calls.

Course Materials

The student kit includes a comprehensive workbook and other necessary materials for this class.

Course Outline

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 1.1: Creating Your First Application

- Exercise 1: Creating Your First Application in Visual Basic .NET

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 2.1: Creating the User Interface

- Exercise 1: Creating the Main Form

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 Message Box 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 3.1: Creating and Using Variables

- Exercise 1: Using Static Variables
- Exercise 2: Creating, Using, and Converting Variables

Lab 3.2: Using Structures and Arrays

- Exercise 1: Creating a Structure
- Exercise 2: Creating and Using 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 4.1: Creating and Using Procedures

- Exercise 1: Creating Functions in a Module
- Exercise 2: Working with the Main Form

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 5.1: Using Decision Structures

- Exercise 1: Checking User Input
- Exercise 2: Confirming Application Close

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 6.1: Validating User Input

- Exercise 1: 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 7.1: Creating a Derived Class

- Exercise 1: Creating a Derived Form 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 8.1: Implementing Structured Exception Handling

- Exercise 1: Using Try…Catch Blocks
- Exercise 2: Using Try…Catch…Finally Blocks

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 10.1: Creating a Web Application

- Exercise 1: Creating the User Interface
- Exercise 2: Using XML Web Services

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 11.1: Accessing Data with ADO.NET

- Exercise 1: Using the Data Form Wizard

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 12.1: Deploying an Application

- Exercise 1: Using the Setup Wizard

- Exercise 2: Installing the Loan 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.