

VB.NET Training

This one-day VB.NET Training course has been designed for programmers wishing to acquire the capability to write sophisticated event-driven applications, or those requiring an understanding of the capabilities of VB.NET. This VB.NET training course uses Visual Studio.NET.

Objective

The aim of this course is to:

- Fully familiarise students with the architecture of the .NET Framework
- Give an understanding of the use of the core classes for building, debugging, and deploying Windows applications.
- Get attendees up to speed on the use of the Visual Studio.NET IDE, including debugging, and application deployment

Details

Duration: 1 Day

Who is this course for

This course is ideal for those who have a familiarity with the Windows operating system and some familiarity with programming constructs. Advanced VB developers should attend a private VB.NET Upgrade course.

If you are unsure if you have the prerequisites for this course, please call 01273 622272 to discuss with one of our training managers.

Course Content

Opening and Running a Visual Basic.NET Program

- The Visual Studio.NET
- Development Environment
- The Visual Studio.NET Tools
- The Windows Forms Designer
- Running a Visual Basic Program
- The Properties Window
- Getting Help
- Writing Your First Program
- Programming Steps
- Creating the User Interface Setting the Properties
- Writing the Code
- Running Visual Basic.NET
- Applications
- Building an Executable File
- Working with Menus and Dialog Boxes

Visual Basic.NET Variables and Operators

- The Anatomy of a Visual Basic
- Program Statement
- Using Variables to Store Information
- Setting Aside Space for Variables
- Using Variables in a Program
- Using a Variable to Store Input
- Using a Variable for Output
- Working with Specific Data Types
- Constants
- Working with Visual Basic
- Operators
- Operator Precedence

Using Decision Structures

- Event-Driven Programming
- Using Conditional Expressions
- If...Then Decision Structures
- Using Logical Operators in
- Conditional Expressions
- Short-Circuiting by Using AndAlso and OrElse
- Select Case Decision Structures
- Using Comparison Operators with a Select Case Structure

Using Loops and Timers

- Writing For...Next Loops
- Creating Complex For...Next Loops
- Writing Do Loops
- Avoiding an Endless Loop
- The Timer Control

Trapping Errors Using Structured Error Handling

- Processing Errors Using Try...Catch
- When to Use Error Handlers
- Path and Disk Drive Errors
- Using the Finally Clause to Perform
- Cleanup Tasks More Complex
- Try...Catch Error Handlers
- The Err Object
- Specifying a Retry Period
- Using Nested Try...Catch Blocks
- Comparing Error Handlers with Defensive Programming Techniques

Using Modules and Procedures

- Working with Standard Modules
- Creating a Standard Module
- Working with Public Variables
- Creating Procedures
- Writing Function Procedures
- Writing Sub Procedures
- Passing Arguments by Value and by Reference

Using Arrays and Collections to Manage Data

- Working with Arrays of Variables
- Creating an Array
- Declaring a Fixed-Size Array
- Working with Array Elements
- Creating a Dynamic Array
- Preserving Array Contents by Using ReDim Preserve
- Working with Object Collections
- Referencing Objects in a Collection
- Writing For Each...Next Loops
- Experimenting with Objects in the Controls Collection

Deploying Visual Basic.NET Applications

- Planning a Deployment
- Different Ways to Deploy an Application
- Creating a Deployment Project
- Customizing Your Deployment Options
- Building a Deployment Project and Testing Setup

Managing Windows Forms

- Adding New Forms to a Program
- How Forms Are Used
- Working with Multiple Forms
- Positioning Forms on the Windows Desktop
- Minimising, Maximising, and Restoring Windows
- Adding Controls to a Form at Runtime
- Organizing Controls on a Form
- Specifying the Startup Object

