

Programming with Visual Basic 2015 for Experienced Programmers

Course ISI-1326B Five Days Instructor-led, Hands on

Introduction

This course is designed to teach experienced programmers how to build business applications with Visual Basic 2015. It is appropriate to those who are new to Visual Basic development environment or if you're upgrading to Visual Basic 2015.

At Course Completion

After completing this course, students will be able to understand all the new language features that make Visual Basic such as outstanding development tool:

- It shows you how to get the most from Visual Studio 2015 as you design, code, debug, and deploy Windows Forms applications using VB 2015.
- It gives you solid training in the essential skills that you'll use every day, including data validation...handling numeric, date, and string data...working with arrays and collections...handling exceptions...working with text, binary, and XML files...and understanding how OOP works.
- It gives you training on database programming.
- It introduces you to LINQ, the .NET feature lets you use VB to query almost any data source, from databases to arrays to XML files.
- It shows you how to build real-world applications, so you always see how the features you're learning interact with each other.

Prerequisites

Before attending this course, students should have the following skills and knowledge:

- Object-Oriented programming experience in a high-level language
- This course is appropriate for students with experience with a previous version of Visual Basic

Student Materials

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

Course Outline

Module 1: An Introduction to Visual Studio

This module introduces you to the basics of working with Visual Basic 2015, the .NET Framework, and Visual Studio. This is the background that you need for getting started right with Visual Basic 2015

- An introduction to .NET development
 - .NET applications
 - Visual Studio and the .NET programming languages
 - The .NET Framework
 - The Visual Studio IDE
 - How a Visual Basic application is compiled and run
 - How Visual Basic differs from the other .NET languages
- A tour of the Visual Studio IDE
 - How to start Visual Studio
 - How to open or close an existing project
 - Some possible menu variations
 - How to use the Form Designer
 - How to use the Code Editor
 - How to use the Solution Explorer
 - How to work with Visual Studio's windows
- How to test a project
- How to build a project
- How to run a project
- How to upgrade projects and change .NET Framework versions
 - How to upgrade projects created in earlier versions of Visual Basic .NET
 - How to change the .NET Framework version used by a project

Module 2: How to design a Windows Forms application

This chapter shows you how to use Visual Studio to design a Windows form that contains labels, text boxes, and buttons. This form will get user input, do two calculations on it, and display the results, so it's not just a "Hello World!" application. When you finish this module, you'll be ready to learn how to write the Visual Basic code for the form.

- How to set options and create a new project
 - How to set the Visual Studio options
 - How to change the environment settings
 - How to create a new project
- How to design a form
 - The design of the Invoice Total form
 - How to add controls to a form
 - How to set properties
 - Common properties for forms and controls
 - How to add navigation features

- The property settings for the Invoice Total form
 - How to use Document Outline view
- How to name and save the files of a project
 - How to name the files of a project
 - How to save the files of a project

Module 3: How to code and test a Windows Forms application

This module shows you how to write the Visual Basic code for the Windows form that was designed in Module 1. It also gets you started with practical debugging techniques.

- An introduction to coding
 - Introduction to object-oriented programming
 - How to refer to properties, methods, and events
 - How an application responds to events
- How to add code to a form
 - How to create an event handler for the default event of a form or control
 - How IntelliSense helps you enter the code for a form
 - Coding rules for Visual Basic statements
 - The event handlers for the Invoice Total form
 - How to code with a readable style
 - How to code comments
 - How to detect and correct syntax errors
- Other skills for working with code
 - How to use the toolbar buttons
 - How to collapse or expand code
 - How to zoom in and out
 - How to print the source code
 - How to code snippets
 - How to rename identifiers
 - How to use the Smart Compile Auto Correction feature
 - How to use the My feature
 - How to get help information
- How to run, test, and debug a project
 - How to run a project
 - How to test a project
 - How to debug runtime errors

Module 4: How to work with numeric and string data

This module shows you how to perform arithmetic operations on various types of numeric data and how to work with string data. Beyond that, this module presents three important subjects that are usually treated too lightly or too late in other books: casting, data conversion, and scope.

- How to work with the built-in value types
 - The built-in value types

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

- How to declare and initialize variables
- How to declare and initialize constants
- How to code arithmetic expressions
- How to code assignment statements
- How to work with the order of precedence
- How to use casting
- How to change the type semantics
- How to use the Math class
- How to work with strings
 - How to declare and initialize a string
 - How to join and append strings
- How to convert data types
 - The .NET structures and classes that define data types
 - How to use Visual Basic functions to convert data types
 - How to use methods of the .NET data structures to convert data types
 - How to use methods of the Convert class to convert data types
 - How to use three of the formatting functions
 - How to use methods to convert numbers to formatted strings
- Three other skills for working with data
 - How to work with scope
 - How to declare and use enumerations
 - How to work with nullable types
- Two versions of the Invoice Total application
 - The basic Invoice Total application
 - The enhanced Invoice Total application

Module 5: How to code control structures

This module shows you how to control program execution and repetitive processing by using conditional statements and various types of loops. Then, it shows you some special debugging techniques for programs with loops in the context of a new application.

- How to code Boolean expressions
 - How to use the relational operators
 - How to use the logical operators
- How to code conditional statements
 - How to code If statements
 - How to code Select Case statements
 - An enhanced version of the Invoice Total application
- How to code loops
 - How to code For loops
 - How to code Do loops
 - How to use Exit and Continue statements
 - Debugging techniques for programs with loops
- The Future Value application
 - The design and property settings for the form

- The code for the form

Module 6: How to code procedures and event handlers

This module shows you how to code and call procedures and how to generate event handlers for all types of events. With those skills, you'll be able to logically divide your code into smaller procedures so it is easier to read, test, and debug.

- How to code and call procedures
 - How to code Sub procedures
 - How to call Sub procedures
 - When and how to pass arguments by reference and by value
 - How to code and call Function procedures
 - How to use optional parameters
- How to work with events
 - How to start an event handler for any event
 - How to handle multiple events with one event handler
 - How to use the Code Editor to start an event handler
 - How to add and remove event wiring
- Another version of the Future Value application
 - The Function procedure
 - The event handlers

Module 7: How to handle exceptions and validate data

First, this module shows you how to use structured exception handling so you can create bulletproof applications. Then, it shows you the best ways to validate the data that the user enters so exceptions are avoided.

- An introduction to data validation and exception handling
 - How to use the isNumeric function for data validation
 - How to display a dialog box for error messages
 - How exception handling works
- How to use structured exception handling
 - How to catch an exception
 - How to use the properties and methods of an exception
 - How to catch specific types of exceptions
 - How to throw an exception
 - The Future Value application with exception handling
- How to validate data
 - How to validate a single entry
 - How to use generic procedures to validate an entry
 - How to validate multiple entries
- The Future Value application with data validation
 - The dialog boxes
 - The code
- Two other ways to validate data

- How to use the Validating event
- How to use the masked text box

Module 8: How to work with arrays and collections

First, this module shows you how to use three types of arrays. Then, it shows you how to use the methods and properties of the Array class for working with an array. Last, it shows you how to use collection classes to work with typed and untyped collections.

- How to work with one-dimensional arrays
 - How to create an array
 - How to assign values to the elements of an array
 - How to use For loops to work with arrays
 - How to use For loops to work with arrays
 - How to use For Each loops to work with arrays
- How to work with rectangular arrays
 - How to create a rectangular array
 - How to assign values to a rectangular array
 - How to work with rectangular arrays
- How to work with jagged arrays
 - How to create a jagged array
 - How to assign values to a jagged array
 - How to work with jagged arrays
- More skills for working with arrays
 - How to use the Array class
 - How to refer to and copy arrays
 - How to code procedures that work with arrays
- How to work with collections
 - Commonly used collection classes
 - Typed vs. untyped collections
 - How to work with a list
 - How to work with a sorted list
 - How to work with queues and stacks
 - How to work with an array list

Module 9: How to work with dates and strings

First, this module shows you how to use the properties and methods of the DateTime structure to work with dates and times. Then, this module shows you how to use the properties and methods of the String and StringBuilder classes to manipulate the data in strings, to validate user entries, and to format numbers, dates, and times.

- How to work with dates and times
 - How to create a DateTime value
 - How to get the current date and time
 - How to format DateTime values
 - How to get information about dates and times

- How to perform operations on dates and times
- How to use Visual Basic properties and functions to work with dates and times
- How to work with strings
 - The properties and methods of the String class
 - Code examples that work with strings
 - More examples that work with strings
 - Two procedures for validating user entries
 - How to use the StringBuilder class
 - How to use the Visual Basic functions to work with strings
- How to format numbers, dates, and times
 - How to format numbers
 - How to format dates and times

Module 10: More skills for working with Windows Forms and Controls

To keep the emphasis on the Visual Basic essentials, the first 9 modules present single-form applications that contain label, text box, and button controls. This module shows you how to develop multi-form applications using controls like combo boxes, list boxes, and check boxes.

- How to work with controls
 - Five more types of controls
 - How to work with combo boxes and list boxes
 - How to work with check boxes and radio buttons
 - How to work with group boxes
 - How to use Tab Order view to set the tab order
 - How to get the information you need for using a control
- How to work with multi-form projects
 - How to add a form to a project
 - How to rename a form
 - How to change the startup form for a project
 - How to display a form as a dialog box
 - How to pass data between a form and a custom dialog box
- How to use the MessageBox class
 - How to display a dialog box and get the user response
 - How to use the FormClosing event
- The Payment application
 - The operation
 - The property settings
 - The code for the Customer form
 - The code for the Payment form

Module 11: How to create and use Classes

This module shows you how to create your own Visual Basic classes that include fields, properties, methods, constructors, and shared members. You will use real-world business objects such as products and customers.

In fact, this module presents an application that uses a Product class to create Product objects and a ProductDB class that uses shared methods to handle the application's database access. That's the type of complete application that you need to see if you want to understand object-oriented programming in Visual Basic.

- An introduction to classes
 - How classes can be used to structure an application
 - The members you can define within a class
 - The code for the Product class
 - How instantiation works
- How to create a class
 - How to add a class file to a project
 - How to code fields
 - How to code standard properties
 - How to code auto-implemented properties
 - How to code methods
 - How to code constructors
 - How to code shared members
 - How to generate code stubs
- The Product Maintenance application
 - The operation of the Product Maintenance application
 - The classes used by the Product Maintenance application
 - The code for the Product Maintenance application
- How to browse classes and use class diagrams
 - How to browse the classes in a solution
 - How to use class diagrams and the Class Details window
- How to work with structures
 - How to create a structure
 - How to use a structure

Module 12: How to Debug an Application

This module presents the most useful debugging features that Visual Studio has to offer. When you finish this module, you'll have all the skills you need for debugging complex applications.

- Basic debugging techniques
 - How to set the debugging options
 - How to work in break mode
 - How to use the Edit and Continue feature
 - How to work with data tips

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

- How to use breakpoints
- How to control the execution of an application
- How to use the debugging windows
 - How to use the Locals window to monitor variables
 - How to use the Autos window to monitor variables
 - How to use Watch windows to monitor expressions
 - How to use the Immediate window to execute commands
 - How to use the Call Stack window to monitor called procedures
 - How to use the Call Hierarchy window to navigate through your code
 - How to use the Output or immediate window to get build or debugging information
 - How to use the Visualizer dialog boxes to view strings

Module 13: An Introduction to Database Programming

Database processing is an integral part of Visual Basic business applications. This module starts by introducing you to database design and SQL queries. Then, it gives you the conceptual background that you need for learning how to develop database applications in the modules that follow.

- An introduction to client/server systems
 - The hardware components of a client/server system
 - The software components of a client/server system
- An introduction to relational databases
 - How a table is organized
 - How the tables in a database are related
 - How the columns in a table are defined
 - The design of the MMABooks database
- How to use SQL to work with a relational database
 - How to query a single table
 - How to join data from two or more tables
 - How to add, update, and delete data in a table
- An introduction to ADO.NET
 - The .NET data providers
 - How the basic ADO.NET components work
 - Concurrency and the disconnected data architecture
 - How a dataset is organized
 - How to work with data without using a data adapter
 - Two ways to create ADO.NET objects

Module 14: How to work with data sources and datasets

Visual Studio 2015 has powerful tools for rapid application development and prototyping of database applications. So in this module, you'll get started with database programming by learning how to use the data sources feature to create applications that let you view and maintain the database data.

- How to create a data source
 - How to use the Data Sources window
 - How to start the Data Source Configuration Wizard
 - How to choose a data source type
 - How to choose the database model for a data source
 - How to choose the connection for a data source
 - How to create a connection to a database
 - How to save a connection string in the App.config file
 - How to choose database objects for a data source
 - The schema file created by the Data Source Configuration Wizard
- How to use a data source
 - How to generate a DataGridView control from a data source
 - A Product Maintenance application that uses a DataGridView control
 - How to change the controls associated with a data source
 - How to generate detail controls from a data source
 - A Customer Maintenance application that uses TextBox controls
- How to handle data errors
 - How to handle data provider errors
 - How to handle ADO.NET errors
 - How to handle data errors for a DataGridView control
- How to use the Dataset Designer
 - How to view the schema for a dataset
 - How to use the Query Builder
 - How to preview the data for a query
 - How to interpret the generated SQL statements

Module 15: How to work with bound controls and parameterized queries

This module builds on what was presented in module 14 to show you how to work with bound controls, use parameterized queries, customize the generated toolbars, and work with a DataGridView control in a database application.

- How to work with bound text boxes and combo boxes
 - How to format the data displayed in a text box
 - How to bind a combo box to a data source
 - How to use code to work with a binding source
- How to work with parameterized queries
 - How to create a parameterized query
 - How to use code to work with a parameterized query
- How to work with the ToolStrip control
 - How to use the Items Collection Editor
 - How to code an event handler for a ToolStrip item
- An enhanced Customer Maintenance application
 - The user interface
 - The code

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

- How to work with a DataGridView control
 - How to modify the properties of a DataGridView control
 - How to edit the columns of a DataGridView control
 - How to format the data in the columns of a DataGridView control
 - How to use a DataGridView control to create a Master/Detail form
- A Customer Invoice Display application
 - The user interface
 - The dataset schema
 - The code for the Customer Invoices form

Module 16: How to use ADO.NET to write your own data access code

Modules 14 and 15 show you how to use data sources to develop database applications. When you do that, Visual Studio generates the ADO.NET objects you need. Now, this module shows you how to create and work with ADO.NET objects through code. That lets you separate the data access code from the presentation code by placing the ADO.NET code in database classes, which are often reusable from one application to another.

- How to work with connections and commands
 - How to create and work with connections
 - How to create and work with commands
- How to create and work with parameters
 - How to use parameters in SQL statements
 - How to create parameters
 - How to work with parameters
- How to execute commands
 - How to create and work with a data reader
 - How to execute queries that return a single value
 - How to execute action queries
- A Customer Maintenance application that uses commands
 - The user interface
 - The class diagram for the business and database classes
 - The code for the CustomerDB class
 - The code for the StateDB class
 - The code for the MMABooksDB class
 - The code for the Customer Maintenance form
 - The code for the Add/Modify Customer form

Module 17: How to work with default properties, events and operators

Module 11 covers the skills that you need whenever you create a Visual Basic class. Now, this module shows you advanced techniques that allow you to develop more complex classes, like how to create default properties, throw argument exceptions, raise events, and overload operators.

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

- An introduction to the ProductList class
 - The code for a simple ProductList class
 - The specifications for the enhanced ProductList class
- How to work with default properties
 - How to create a default property
 - How to throw an argument exception
- How to work with events
 - How to define events
 - How to use events
 - How to overload operators
- An introduction to operator overloading
 - How to overload arithmetic operators
 - How to overload relational operators
- An enhanced version of the Product Maintenance application
 - The code for the ProductList class
 - The code for the Product Maintenance form

Module 18: How to work with inheritance

This module shows you how inheritance is used throughout the .NET classes and how you can use inheritance in your own VB classes. When you're done, you'll know how to create base and derived classes, how polymorphism works, and how to use casting with inheritance.

To make sure that you get all of that, this module presents a complete application that uses a Product base class, Book and Software derived classes, and a ProductList class that inherits the List() class.

- Introduction to inheritance
 - How inheritance works
 - How the .NET Framework uses inheritance
 - Methods inherited from the System Object class
 - How to use inheritance in your applications
- Basic skills for working with inheritance
 - How to create a base class
 - How to create a derived class
 - How polymorphism works
- An inheritance version of the Product Maintenance application
 - The operation of the Product Maintenance application
 - The code for the Product, Book, and Software classes
 - The code for the ProductList class
 - The code for the Product Maintenance form
 - The code for the New Product form
- Object types and casting
 - How to use the Type class to get information about an object's type
 - How to use casting with inheritance

- How to work with abstract and sealed classes
 - How to work with abstract classes
 - How to work with sealed classes

Module 19: How to work with interfaces and generics

Although a Visual Basic class can inherit only one class, it can implement one or more interfaces. So in this module, you'll learn how to create and implement a Visual Basic interface. You'll also learn how to use generics so you can code your own collections that work like the typed collections from the .NET Framework presented in module 8. Along the way, you'll learn how to work with the generic interfaces that are used with generic collections.

- How to work with interfaces and generics
 - An introduction to interfaces
 - Some of the interfaces defined by the .NET Framework
 - How to create an interface
 - How to implement an interface
 - A Product class that implements the ICloneable interface
 - How to use an interface as a parameter
- How to work with generics
 - How to code a class that defines a generic collection
 - Some of the generic interfaces defined by the .NET Framework
 - How to implement the IComparable() interface
 - How to use constraints
 - How to code an interface that uses generics

Module 20: How to organize and document your classes

This module shows you how to organize the classes you create and document them using XML. It also shows you how to store your classes in class libraries so that other programmers can easily access them.

- How to organize your classes
 - How to code multiple classes in a single file
 - How to split a single class across multiple files
 - How to work with namespaces
- How to document your classes
 - How to add XML documentation to a class
 - How to view the XML documentation
- How to create and use class libraries
 - How class libraries work
 - How to create a class library project
 - How to add a reference to a class library
 - How to use the classes in a class library

Module 21: How to work with files and data streams

Although databases are commonly used in Visual Basic applications, you may also need to access data that's stored in a text file or a binary file. So this module shows you how to read and write both types of files. It also shows you how to use one of the My objects to work with files.

- An introduction to the System.IO classes
 - The classes for managing directories, files, and paths
 - How files and streams work
 - How to use the FileStream class
 - How to use the exception classes for file I/O
 - How to use the MyComputer FileSystem object
- How to work with text files
 - How to write a text file
 - How to read a text file
 - A class that works with a text file
- How to work with binary files
 - How to write a binary file
 - How to read a binary file
 - A class that works with a binary file

Module 22: How to work with XML files

To start, this module presents the basics of XML and shows you how to use the Visual Studio XML Editor for working with XML. Then, it shows you how to use Visual Basic with the XmlWriter and XmlReader classes to store XML documents in a file and to read XML documents from a file.

- An introduction to XML
 - An XML document
 - XML tags, declarations, and comments
 - XML elements
 - XML attributes
- How to work with the XML Editor
 - How to create a new XML file
 - How to open an existing XML file
 - How to edit an XML file
- How to work with XML
 - How to use the XmlWriter class
 - Code that writes an XML document
 - How to use the XmlReader class
 - How the XmlReader class reads nodes
 - Code that reads an XML document
 - A class that works with an XML file

Module 23: How to Use LINQ

In this module, you'll learn the basic skills for using LINQ, a feature that lets you query a data source using constructs that are built into the Visual Basic language. That means you can use the same language to access a variety of data sources, from databases to arrays to XML files.

- Basic concepts for working with LINQ
 - How LINQ is implemented
 - Advantages of using LINQ
 - The three stages of a query operation
- How to code a LINQ query
 - How to identify the data source for a query
 - How to filter the results of a query
 - How to sort the results of a query
 - How to select fields from a query
 - How to join data from two or more data sources
- A Customer Invoice application that uses generic lists
 - The user interface
 - The code for the form
- A Customer Invoice application that uses a typed dataset
 - The dataset schema
 - The code for the form

Module 24: How to enhance the user interface

This module shows you how to add another level of professionalism to your Visual Basic applications by developing single- or multiple-document interfaces that include menus, toolbars, status bars, Main procedures, and help information.

- Two types of user interfaces
 - A single-document interface (SDI)
 - A multiple-document interface (MDI)
- How to develop SDI applications
 - How to use a startup form
 - How to use a Tab control
- How to add menus to a form
 - How to create menus
 - How to set the properties that work with menu items
 - How to write code that works with menu items
- How to develop MDI applications
 - How to create parent and child forms
 - How to write code that works with parent and child forms
- How to add toolbars to a form
 - How to create a toolbar
 - How to write code that works with toolbars
- How to create and use a status bar

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

- How to create a simple status bar
 - How to create a status bar with multiple panels
- How to work with modules
 - How to create and code a module
 - How to use a Main procedure
- How to add help information
 - How to add tool tips
 - How to add context-sensitive help

Module 25: How to deploy an application

As you develop a Visual Basic application, you'll need to deploy it to a target system in order to test it thoroughly. And, of course, once the application is finished, you'll need to deploy it to users' systems. So this module shows you three ways you can deploy applications from a network or web server using Visual Studio.

- An introduction to deploying Windows applications
 - How XCopy works
 - How ClickOnce works
 - How a Setup program works
- How to use XCopy
 - How to create a release build
 - How to copy the release build to the client
- How to use ClickOnce
 - How to publish an application
 - How to select the files that are published
 - How to select the prerequisites
 - How to set the update options
 - How to set the publish options
 - How to install an application
 - How update an application
- How to create and use a Setup program
 - How to create an InstallShield project
 - How to use the InstallShield Project Assistant
 - How to add output files to an InstallShield project
 - How to create and view the installation files for a Setup program
 - How to use a Setup program to install an application
- How to deploy database applications
 - Using ClickOnce deployment
 - Using a Setup program