Programming in Visual Basic with Microsoft Visual Studio 2010

Duration: 5 Days Course Code: M10550

Overview:

This course teaches you Visual Basic language syntax, program structure, and implementation by using Microsoft Visual Studio 2010 and the Microsoft .NET Framework 4.
This course provides a solid foundation in Visual Basic to the level necessary to enable students to attend other courses in the Technical Specialist tracks

Target Audience:

This course is intended for experienced developers who already have programming experience in Visual Basic, C, C++, C#, or Java, and understand the concepts of Object Oriented Programming. These developers will be likely to develop enterprise business solutions.
These professional developers will be attending the course so that they can quickly ramp up on Visual Basic Programming in the .NET Framework. The course focuses on Visual Basic program structure, language syntax, and implementation details with the .NET Framework 4.0. This course also focuses on new enhancement in the Visual Basic 2010 language using Visual Studio 2010.

Objectives:

- After completing this course, students will be able to:
- Describe the purpose of the .NET Framework, and explain how to use Microsoft Visual Basic and Visual Studio 2010 to build .NET Framework applications.
- Describe the syntax of basic Visual Basic programming constructs.
- Describe how to create and call methods.
- Describe how to catch, handle, and throw exceptions.
- Describe how to perform basic file I/O operations in a Visual Basic application.
- Describe how to create and use new types (enumerations, classes, and structures), and explain the differences between reference types and value types.
- Describe how to control the visibility and lifetime of members in a type.
- Describe how to use inheritance to create new reference types.
- Describe how to manage the lifetime of objects and control the use of resources.
- Describe how to create properties and indexers to encapsulate data, and explain how to define operators for this data.
- Describe how to decouple an operation from the method that implements it, and explain how to use these decoupled operations to handle asynchronous events.
- Describe the purpose of collections, and explain how to use generics to implement type-safe collection classes, structures, interfaces, and methods.
- Describe how to implement custom collection classes that support enumeration.
- Describe how to query in-memory data by using Language-Integrated Query (LINQ) queries.
- Describe how to integrate code written by using a dynamic language such as Ruby and Python, or technologies such as Component Object Model (COM), into a Visual Basic application.

Prerequisites:

This course requires that you meet the following prerequisites:

This course is targeted at developers who already have Visual Basic knowledge.

This course is not for new developers; at least 12 months experience working with an Object Oriented language is expected.

- Creating classes

- Inheritance and abstraction
- Polymorphism
- Interfaces
- Delegates
- Events
- Exceptions

Experience with the Microsoft .NET Framework

Knowledge of the Visual Studio integrated development environment (IDE).

Content:

Module 1: Introducing Visual Basic and the .NET Framework

- This module describes the purpose of the .NET Framework 4 and how you can build applications by using Visual Studio 2010.

Module 2: Using Visual Basic Programming Constructs

- This module introduces many of the basic Visual Basic language data types and programming constructs, and describes the syntax and semantics of these constructs.

Module 3: Declaring and Calling Methods

- A key part of developing any application is dividing the solution into logical components. In object-oriented languages such as Microsoft Visual Basic, a method is a unit of code that is designed to perform a discrete piece of

Module 4: Handling Exceptions

- Exception handling is an important concept and your applications should be designed with exception handling in mind. This module explains how you can implement effective exception handling in your applications, and how you can

Module 5: Reading and Writing Files

- The ability to access and manipulate the files on the file system is a common requirement for many applications. This module shows how to read and write to files by using the classes in the Microsoft .NET Framework. This module

Module 6: Creating New Types

- The Microsoft.NET Framework base class library consists of many types that you can use in your applications. However, in all applications, you must also build your own types that implement the logic for your solution.
  This m

Module 7: Encapsulating Data and Methods

- This module describes how to use some of the access modifiers that Visual Basic provides to enable you to implement encapsulation. This module also introduces the Shared modifier, which enables you to define members that can be

Module 8: Inheriting from Classes and Implementing Interfaces

- This module introduces inheritance and interfaces in the Microsoft .NET Framework, and how you can use them to simplify complex problems, reduce code duplication, and speed up development.
  Inheritance is a key concept in an

Module 9: Managing the Lifetime of Objects and Controlling Resources

- All applications use resources. When you build a Microsoft Visual Basic application, resources fall into two broad categories: managed resources that are handled by the common language runtime (CLR) and unmanaged resources that

Module 10: Encapsulating Data and Defining Overloaded Operators

- Many operators have well-defined behavior for the built-in Visual Basic types, but you can also define operators for your own types. This module describes how to implement operators for your types by using overloading.

Module 11: Decoupling Methods and Handling Events

- This module explains how to decouple an operation from the method that implements it and how to use anonymous methods to implement decoupled operations. This module also explains how to use events to inform consuming applicatio

Module 12: Using Collections and Building Generic Types

- The basic collection classes introduce a new problem. Classes that act on other types are often not type-safe. For example, many collection classes frequently use the Object type to store items, and must then be cast or convert

Module 13: Building and Enumerating Custom Collection Classes

- When you develop applications, you often need to store collections of objects. In many circumstances, you can use the collection classes that the Microsoft .NET Framework includes; however, sometimes these collection classes do

Module 14: Using LINQ to Query Data

- This module introduces you to Language-Integrated Query (LINQ) queries and explains how you can use them to process data in your Microsoft .NET Framework applications. This module also explains the difference between shared and

Module 15: Integrating Visual Basic Code with Dynamic Languages and COM Components

- Integration with other technologies is a key feature of the Microsoft.NET Framework. Previous versions of the .NET Framework enabled you to combine components that were developed by using different languages that have compilers

Further Information:

For More information, or to book your course, please call us on 00 20 (0) 2 2269 1982 or 16142

training@globalknowledge.com.eg

www.globalknowledge.com.eg

Global Knowledge, 16 Moustafa Refaat St. Block 1137, Sheraton Buildings, Heliopolis, Cairo