MS10266
Längd: 5 dagar

Programming with C# using .NET Framework 4


Förkunskaper
Kursen lämpar sig både för dig som antingen arbetat med tidigare versioner av C# och vill se förbättringarna i C# 4.0 såväl för dig som arbetat med andra objektorienterade språk och vill gå över till C#. För dig som inte arbetat med objektorienterad utveckling tidigare rekommenderar vi MS10267 som ett första steg.

Språk
Kursen hålls på svenska (Kontakta oss om du föredrar engelska).

Kursmaterial
Microsofts officiella elektroniska kursmaterial på engelska ingår.

Vi rekommenderar att du väljer kursen MS20483 Programming in C# using .NET Framework 4.5 som genomförs på senaste plattformen.

Course Outline:

Module 1: Introducing C# and the .NET Framework
provides an overview of the .NET Framework and shows how you can start to build your own .NET Framework applications by using C# and Visual Studio 2010.

Module 2: Using C# Programming Constructs
provides an introduction to C# programming language syntax and introduces many of the basic C# language data types and programming constructs.

Module 3: Declaring and Calling Methods
introduces the concept of methods and describes how, in object-oriented languages such as C#, a method is a unit of code that is designed to perform a discrete piece of work. This module shows you how to declare and call methods by using C#.

Module 4: Handling Exceptions
introduces the importance of exception handling and explains why applications should be designed with exception handling in mind. This module explains how you can implement effective exception handling in your applications and describes how to use exceptions in your methods to indicate an error condition to the code that calls your methods.

Module 5: Reading and Writing Files
explains how the ability to access and manipulate files on the file system is a common requirement for many applications. This module shows you how to read and write to files by using the classes in the .NET Framework It also describes the different approaches that you can take and explains how to read and write different formats of data.

Module 6: Creating New Types
explains how to build your own types that model items in the real world and describes how to implement the business logic for these items that your applications require. This module explains the differences between reference types and value types.

Module 7: Encapsulating Data and Methods
describes how to use the access modifiers that C# provides to enable you to implement encapsulation. This module also introduces the static modifier, which enables you to define members that can be shared over multiple instances of the same type.
Module 8: Inheriting from Classes and Implementing Interfaces
explains that inheritance is a key concept in an object-oriented language and describes how you can use inheritance, interfaces, and abstract classes to develop object hierarchies. This module also explains how you can use these object hierarchies to help reduce bugs by defining clear contracts for the functionality that a class should expose and providing default implementations where you can sensibly abstract code into a base type.

Module 9: Managing the Lifetime of Objects and Controlling Resources
introduces the concept of resource management and discusses its importance. This module explains how the NET Framework simplifies resource management by automatically reclaiming the resources for a managed object when an application no longer references it. This module also explains that the garbage collector does not control unmanaged resources and describes the steps that you can take to dispose of such resources.

Module 10: Encapsulating Data and Defining Overloaded Operators
introduces properties and indexers. These are elements of C# that enable you to encapsulate data and expose data appropriately and efficiently. This module also describes how to implement operators for your types by using overloading.

Module 11: Decoupling Methods and Handling Events
explains how to decouple an operation from the method that implements it and describes how to use anonymous methods to implement decoupled operations. This module also explains how to use events to inform consuming applications of a change or notable occurrence in a type.

Module 12: Using Collections and Building Generic Types
introduces the concept of collection classes and explains that you can use them with greater flexibility than a simple array. This module also introduces generics and explains how to use generic classes to maintain type integrity and avoid the issues that are associated with a lack of type safety.

Module 13: Building and Enumerating Custom Collection Classes
explains how to use the collection classes that the .NET Framework Base Class Library includes. This module also describes how to build custom collection classes.

Module 14: Using LINQ to Query Data
explains how you can use LINQ to abstract the mechanism that an application uses to query data from the application code. This module describes built-in C# LINQ extension methods and LINQ query operators. This module also describes how to build LINQ queries dynamically by using expression trees.

Module 15: Integrating Visual C# Code with Dynamic Languages and COM Components
Explains how the .NET Framework 4.0 enables you to invoke code and components that were written by using other languages from your C# code. It describes how the Dynamic Language Runtime (DLR) enables you to reuse code built by using a wide range of scripting languages, such as Ruby and Python. This module also describes how to invoke COM components from a C# application.