Programming in C# (55339AC)

Course outline

Module 1: C# Syntax

Microsoft .NET 6 provides a comprehensive development platform that you can use to build, deploy, and manage applications and services. By using .NET, you can create visually compelling applications, enable seamless communication across technology boundaries, and provide support for a wide range of business processes.

In this module, you'll learn about some of the core features provided by.NET and Microsoft Visual Studio. You'll also learn about some of the core C# constructs that enable you to start developing .NET applications.

Lessons

- Lesson 1: Writing Applications in C# and .NET
- Lesson 2: Types of Data and Expressions
- Lesson 3: C# Language Constructs

Lab 1: Developing the Class Enrolment Application

- Developing the Class Enrolment Application

After completing this module, students will be able to:

- Write Applications in C# and .NET
- Explain types of Data and Expressions
- Understand C# Language Constructs

Module 2: C# Language Concepts

Applications often consist of logical units of functionality that perform specific functions, such as providing access to data or triggering some logical processing. C# is an object-orientated language and uses the concept of methods to encapsulate logical units of functionality. Although a good practice is to have methods that do just one thing, they can be as simple or as complex as you like. It is also important to consider what happens to the state of your application when an exception occurs in a method.

Lessons

- Lesson 1: Methods
- Lesson 2: Method Overloading
- Lesson 3: Exception Handling
- Lesson 4: Monitoring

Lab 1: Extending the Class Enrolment Application

- Refactor code to facilitate reusability.
- Write C# code that validates data entered by a user.
- Write C# code that saves changes back to a database.

After completing this module, students will be able to:

- In this module, you'll learn how to create and use methods and how to handle exceptions. You'll also learn how to use logging and tracing to record the details of any exceptions that occur

Module 3: C# Structures, Collections and Events

To create effective applications you must first learn some fundamental C# constructs. You need to know how to create simple structures to represent the data items you are working with. You need to know how to organize these structures into collections, so that you can add items, retrieve items, and iterate over your items. Finally, you need to know how to subscribe to events so that you can respond to the actions of your users.

Lessons

- Lesson 1: Structs
- Lesson 2: Enums
- Lesson 3: Built-in Collections
- Lesson 4: Events

Lab 1: Building the Grades Prototype Application

- Structs
- Enums
- Built-in Collections
- Events

After completing this module, students will be able to:

- Create and use structs and enums
- Organize data into collections
- Create and subscribe to events

Module 4: C# Classes

In this module, you'll learn how to use interfaces and classes to define and create your own custom, reusable types. You'll also learn how to create and use enumerable type-safe collections of any type.

Lessons

- Lesson 1: Creating Classes
- Lesson 2: Interfaces
- Lesson 3: Understanding Generics in C#

Lab 1: Adding Data Validation to the Application

- Creating Classes
- Interfaces
- Understanding Generics in C#

After completing this module, students will be able to:

- Use interfaces and classes to define and create custom, reusable types
- Create and use enumerable type-safe collections of any type

Module 5: C# Inheritance
In this module, you'll learn how to use inheritance to create class hierarchies and to extend .NET types.

Lessons

- Lesson 1: Hierarchies of Classes
- Lesson 2: Polymorphism
- Lesson 3: Extending Classes

Lab 1: Refactoring

- Hierarchies of Classes
- Polymorphism
- Extending Classes

After completing this module, students will be able to:

- Use inheritance to factor common functionality into a base class.
- Implement polymorphism by using an abstract method.
- Create a custom exception class.

Module 6: Input and Output
In this module, you'll learn how to read and write data by using transactional filesystem I/O operations, how to serialize and deserialize data to the filesystem, and how to read and write data to the filesystem by using streams.

Lessons

- Lesson 1: File I/O
- Lesson 2: Serialization and Deserialization
- Lesson 3: Streams

Lab 1: Creating the Grades Report

- File I/O
- Serialization and Deserialization
- Streams

After completing this module, students will be able to:

- Read and write data by using transaction filesystem I/O operations
- How to searlize and deserialize data to the file system
- How to read and write data to the filesystem by using streams.

Module 7: Database Access
In this module, you'll learn how to use Entity Framework and how to query many types of data by using Language-Integrated Query (LINQ).

Lessons

- Lesson 1: Entity Framework
- Lesson 2: LINQ

Lab 1: Updating Grade Data

- Entity Framework
- LINQ

After completing this module, students will be able to:

- Use entity Framework
- Learn how to query many types of data by using Language-Integrated Query (LINQ).

Module 8: Using the Network

In this module, you'll learn how to use the request and response classes in the System.Net namespace to directly manipulate remote data sources. You'll also learn about REST and OData and look briefly at ASP.NET Core MVC.

Lessons

- Lesson 1: Web Services
- Lesson 2: REST and OData
- Lesson 3: ASP.NET Core MVC

Lab 1: None

- None

After completing this module, students will be able to:

- Send data to remote web services.
- Access remote data over web services.
- Understand REST and OData.

Module 9: Graphical User Interfaces

In this module, you'll learn how to use Extensible Application Markup Language (XAML) and Windows Presentation Foundation (WPF) to create engaging UIs.

Lessons

- Lesson 1: Using UI Frameworks
- Lesson 2: Data binding
- Lesson 3: Styling the UI

Lab 1: Adding a Graphical User Interface

- Using UI Frameworks
- Data binding
- Styling the UI

After completing this module, students will be able to:

- Use Extensible Application Markup Language (XAML)
- Create and use user controls.
- Use styles and animations.

Module 10: Application Performance
In this module, you'll learn how to improve the performance of your applications by distributing your operations across multiple threads.

Lessons

- Lesson 1: Multitasking
- Lesson 2: Asynchronous Calls
- Lesson 3: Dealing with Conflicts

Lab 1: Performance Tuning

- Multitasking
- Asynchronous Calls
- Dealing with Conflicts

After completing this module, students will be able to:

- Improve performance by distributing operations across multiple threads.
- Use the async and await keywords to implement asynchronous methods.
- Use events and user controls to provide visual feedback during long-running operations.

Module 11: C# Interop
In this module, you'll learn how to interoperate with unmanaged code in your applications and how to ensure that your code releases any unmanaged resources.

Lessons

- Lesson 1: Dynamic Objects
- Lesson 2: Managing Resources

Lab 1: Working with Word

- Dynamic Objects
- Managing Resources

After completing this module, students will be able to:

- Interoperate with unmanaged code in applications.
- Ensure that code releases any unmanaged resources.

Module 12: Designing for Reuse
In this module, you'll learn how to consume existing assemblies by using reflection, and how to add additional metadata to types and type members by using attributes. You'll also learn how to generate code at runtime by using the Code Document Object Model (CodeDOM) and how manage your .NET assemblies.

Lessons

- Lesson 1: Metadata
- Lesson 2: Attributes
- Lesson 3: Generating Code
- Lesson 4: Assemblies

Lab 1: Managing the Grades Report Assembly

- Metadata
- Attributes
- Generating Code
- Assemblies

After completing this module, students will be able to:

- Consume existing assemblies by using .NET reflection.
- Add additional metadata to types and type members using attributes.
- Create custom attributes.
- Get information about assemblies.