

KOENIG SOLUTIONS · COURSE OUTLINE

Programming in C#

Learn to build .NET applications with C#, from core syntax and classes to databases, UIs, threading and encryption.

13

MODULES

35

TOPICS

49

HANDS-ON LABS

COURSE OVERVIEW

This course covers core Visual C# syntax, methods, exception handling, and application monitoring, followed by structs, enums, collections, and events. It then addresses creating classes, interfaces, type-safe collections, and inheritance hierarchies. Further modules cover reading and writing local data, accessing databases with entity data models and LINQ, and accessing remote data through web services. The course concludes with designing user interfaces with XAML and WPF, improving performance through multitasking, integrating unmanaged code, building reusable assemblies, and encrypting and decrypting data.

WHO THIS COURSE IS FOR

C# Developers

Software Engineers

.NET Developers

Application Developers

Database Developers

LEARNING OBJECTIVES

- Apply core Visual C# syntax, data types, and operators
- Create methods and handle exceptions in applications
- Implement structs, enums, collections, and events
- Create classes, interfaces, and type-safe collections
- Build class hierarchies using inheritance
- Read and write local data using files and streams
- Query databases using entity data models and LINQ
- Design user interfaces with XAML and WPF

13 Modules, 35 Topics

01

4 Labs

Review of Visual C# Syntax

The Microsoft .NET Framework version 4.7 provides a comprehensive development platform that you can use to build, deploy, and manage applications and services. By using the .NET Framework, 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 will learn about some of the core features provided by the .NET Framework and Microsoft Visual Studio. You will also learn about some of the core Visual C# constructs that enable you to start developing .NET Framework applications.

TOPICS

- Overview of Writing Application by Using Visual C#
- Data Types, Operators, and Expressions
- Visual C# Programming Language Constructs

02

4 Labs

Creating Methods, Handling Exceptions, and Monitoring Applications

Applications often consist of logical units of functionality that perform specific functions, such as providing access to data or triggering some logical processing. Visual C# is an object-orientated language and uses the concept of methods to encapsulate logical units of functionality. A method can be as simple or as complex as you like, and therefore it is important to consider what happens to the state of your application when an exception occurs in a method. In this module, you will learn how to create and use methods and how to handle exceptions. You will also learn how to use logging and tracing to record the details of any exceptions that occur.

TOPICS

- Creating and Invoking Methods
- Creating Overloaded Methods and Using Optional and Output Parameters
- Handling Exceptions
- Monitoring Applications

03**4 Labs**

Basic Types and Constructs of Visual C#

To create effective applications by using Windows Presentation Foundation (WPF) or other .NET Framework platforms, you must first learn some basic Visual 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. In this module, you will learn how to create and use structs and enums, organize data into collections, and create and subscribe to events.

TOPICS

Implementing Structs and Enums · Organizing Data into Collections · Handling Events

04**5 Labs**

Creating Classes and Implementing Type-Safe Collections

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

TOPICS

Creating Classes · Defining and Implementing Interfaces · Implementing Type-Safe Collections

05**3 Labs**

Creating a Class Hierarchy by Using Inheritance

In this module, you will learn how to use inheritance to create class hierarchies and to extend .NET Framework types.

TOPICS

Creating Class Hierarchies · Extending .NET Framework Classes

06**4 Labs**

Reading and Writing Local Data

In this module, you will learn how to read and write data by using transactional file system I/O operations, how to serialize and deserialize data to the file system, and how to read and write data to the file system by using streams.

TOPICS

Reading and Writing Files · Serializing and Deserializing Data · Performing I/O by Using Streams

07

4 Labs

Accessing a Database

In this module, you will learn how to create and use entity data models (EDMs) and how to query many types of data by using Language-Integrated Query (LINQ).

TOPICS

Creating and Using Entity Data Models · Querying Data by Using LINQ

08

4 Labs

Accessing Remote Data

In this module, you will learn how to use the request and response classes in the System.Net namespace to directly manipulate remote data sources. You will also learn how to use Windows Communication Foundation (WCF) Data Services to expose and consume an entity data model (EDM) over the web.

TOPICS

- Accessing Data Across the Web
- Accessing Data by Using OData Connected Services

09

4 Labs

Designing the User Interface for a Graphical Application

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

TOPICS

Using XAML to Design a User Interface · Binding Controls to Data

10

3 Labs

Improving Application Performance and Responsiveness

In this module, you will learn how to improve the performance of your applications by distributing your operations across multiple threads.

TOPICS

Implementing Multitasking · Performing Operations Asynchronously · Synchronizing Concurrent Access to Data

11

3 Labs

Integrating with Unmanaged Code

In this module, you will learn how to interoperate unmanaged code in your applications and how to ensure that your code releases any unmanaged resources.

TOPICS

- Creating and Using Dynamic Objects
- Managing the Lifetime of Objects and Controlling Unmanaged Resources

12

4 Labs

Creating Reusable Types and Assemblies

In this module, you will learn how to consume existing assemblies by using reflection and how to add additional metadata to types and type members by using attributes. You will also learn how to generate code at run time by using the Code Document Object Model (CodeDOM) and how to ensure that your assemblies are signed and versioned, and available to other applications, by using the global assembly cache (GAC).

TOPICS

Examining Object Metadata · Creating and Using Custom Attributes · Generating Managed Code · Versioning, Signing, and Deploying Assemblies

13

3 Labs

Encrypting and Decrypting Data

In this module, you will learn how to implement symmetric and asymmetric encryption and how to use hashes to generate mathematical representations of your data. You will also learn how to create and manage X509 certificates and how to use them in the asymmetric encryption process.

TOPICS

Implementing Symmetric Encryption · Implementing Asymmetric Encryption

Practise What You Learn

#	MODULE	ACTIVITY
01	Review of Visual C# Syntax	■ Implementing Edit Functionality for the Students List ■ Implementing Insert Functionality for the Students List ■ Implementing Delete Functionality for the Students List ■ Displaying a Student's Age
02	Creating Methods, Handling Exceptions, and Monitoring Applications	■ Extending the Class Enrolment Application Functionality ■ Refactoring the Enrolment Code ■ Validating Student Information ■ Saving Changes to the Class List
03	Basic Types and Constructs of Visual C#	■ Writing the Code for the Grades Prototype Application ■ Adding Navigation Logic to the Grades Prototype Application ■ Creating Data Types to Store User and Grade Information ■ Displaying User and Grade Information
04	Creating Classes and Implementing Type-Safe Collections	■ Adding Data Validation and Type-Safety to the Application ■ Implementing the Teacher, Student, and Grade Structs as Classes ■ Adding Data Validation to the Grade Class ■ Displaying Students in Name Order ■ Enabling Teachers to Modify Class and Grade Data
05	Creating a Class Hierarchy by Using Inheritance	■ Refactoring Common Functionality into the User Class ■ Implementing Password Complexity by Using an Abstract Method ■ Creating the ClassFullException Custom Exception
06	Reading and Writing Local Data	■ Generating the Grades Report ■ Serializing Data for the Grades Report as XML ■ Previewing the Grades Report ■ Persisting the Serialized Grade Data to a File
07	Accessing a Database	■ Retrieving and Modifying Grade Data ■ Creating an Entity Data Model from The School of Fine Arts Database ■ Updating Student and Grade Data by Using the Entity Framework ■ Extending the Entity Data Model to Validate Data
08	Accessing Remote Data	■ Retrieving and Modifying Grade Data Remotely ■ Creating a WCF Data Service for the SchoolGrades Database ■ Integrating the Data Service into the Application ■ Retrieving Student Photographs Over the Web (If Time Permits)

#	MODULE	ACTIVITY
09	Designing the User Interface for a Graphical Application	■ Customizing Student Photographs and Styling the Application ■ Customizing the Appearance of Student Photographs ■ Styling the Logon View ■ Animating the StudentPhoto Control (If Time Permits)
10	Improving Application Performance and Responsiveness	■ Improving the Responsiveness and Performance of the Application ■ Ensuring That the UI Remains Responsive When Retrieving Teacher Data ■ Providing Visual Feedback During Long-Running Operations
11	Integrating with Unmanaged Code	■ Upgrading the Grades Report ■ Generating the Grades Report by Using Word ■ Controlling the Lifetime of Word Objects by Implementing the Dispose Pattern
12	Creating Reusable Types and Assemblies	■ Specifying the Data to Include in the Grades Report ■ Creating and Applying the IncludeInReport attribute ■ Updating the Report ■ Storing the Grades.Utilities Assembly Centrally (If Time Permits)
13	Encrypting and Decrypting Data	■ Encrypting and Decrypting the Grades Report ■ Encrypting the Grades Report ■ Encrypting the Grades Report

WHAT IS INCLUDED

Everything You Need to Succeed

- Koenig course material for every module
- Delivery by certified instructors and subject matter experts
- Hands-on exercises throughout the course
- Course completion certificate from Koenig

ABOUT KOENIG

Who We Are

Established in 1993, Koenig Solutions is a globally recognised training provider with 35+ technology partnerships, 5,000+ training programmes and a network of certified instructors and subject matter experts, supporting individuals and enterprise teams worldwide.

FLEXIBLE DELIVERY

Learn the Way That Suits You

Classroom Training

Live Online Classes (ILO)

Fly-Me-A-Trainer (FMAT)

Flexi (Self-Paced)

1-on-1 Training

Ready to Get Started?

Talk to a learning advisor about schedules, group training and customised delivery.

Visit koenig-solutions.com

✉ info@koenig-solutions.com