Faysal Aziz

Programming in C#

3-day outline for beginners

FA-0367 · 3-day

Software Development

Introduction

This program is meticulously designed to provide an immersive learning experience, combining theoretical knowledge with practical skills in C#. We're proud to inform participants that our trainers are industry veterans with extensive experience. They have worked with top Research and Development bodies internationally, contributing to innovative projects, pioneering research, and cutting-edge technology in software development. Their real-world experience not only enriches the course content with practical insights but also ensures that the knowledge imparted is up-to-date with current industry standards and practices.

By leveraging their backgrounds, our trainers have tailored the course to simulate real-world software development scenarios, helping participants to see beyond the code and understand the broader context of software development in a professional environment.

Learning outcomes

Upon successful completion of this course, participants will be able to:

- Understand the principles of C# programming, including the concepts that underline this object-oriented language.
- Grasp the syntax and structure of C# to create clean and efficient code.
- Develop robust applications using C#'s standard class libraries.
- Handle errors and exceptions effectively, ensuring software reliability and resilience.
- Implement file operations, memory management, and various I/O principles in C#.
- Utilize LINQ for efficient data querying and manipulation.
- Understand and implement advanced concepts such as asynchronous programming, threading, and task parallelism.
- Design and develop basic APIs with proper authentication.

- Test, debug, and deploy C# applications, implementing best practices.
- Understand the integration of C# in the .NET framework and how it interoperates with different technologies.
- Build a foundational understanding of how C# can be used in various software development paradigms, including web applications, desktop applications, and mobile development.

Prerequisites

Before enrolling in this course, participants should have:

- Basic knowledge of computer operations.
- Understanding of programming principles and experience in at least one programming language (preferably).
- Familiarity with the concepts of object-oriented programming (OOP).
- Basic understanding of common data structures and algorithms.
- Access to a personal computer or laptop with the necessary software installed for C# development (preferably Visual Studio).
- A proactive mindset and willingness to participate in interactive sessions, including group activities and discussions.

Training outline

1. Introduction to C#:

1. Overview of C#, understanding the environment setup.
2. Discussing the history and evolution of C#.
3. Exploring .NET Framework and .NET Core.
4. Fundamentals of C# programming.

2. Diving into C# Basics:

1. Understanding variables, data types, and constants.
2. Control structures: loops, conditionals, and decision-making constructs.
3. Organizing code with namespaces, assemblies, and applications.

3. Object-Oriented Programming in C#:

1. Defining classes and objects.
2. Implementing encapsulation, inheritance, and polymorphism.

3. Exploring methods, properties, events, and indexers.
4. Understanding object life cycle and garbage collection.
5. Practical OOP: designing a simple system to implement OOP concepts.

4. Advanced C# Concepts:

1. Delegates and events: understanding their necessity and use cases.
2. Anonymous methods, lambda expressions, and closures.
3. Exploring iterators, yield, and managing collections with generics.
4. Dive into asynchronous programming: async, await, tasks, and parallel programming.

5. Data Handling and LINQ:

1. Overview of data management in C#.
2. Introduction to LINQ, understanding its advantages.
3. Practical uses of LINQ in data querying, filtering, and manipulation.
4. Entity Framework basics: Object Relational Mapping (ORM) concept.

6. File I/O and Serialization:

1. Managing the File System (reading/writing files, file system operations).
2. Understanding serialization and deserialization in C#.
3. Practical examples: handling different file formats (e.g., JSON, XML).

7. Error and Exception Handling:

1. Basics of error handling: exceptions and try-catch blocks.
2. Custom exceptions: creating and throwing them.
3. Using finally and using blocks for effective resource management.

8. Testing and Debugging:

1. Writing test cases and understanding unit testing in C#.
2. Effective debugging techniques.
3. Using diagnostic tools within the development environment.

9. Building APIs and Web Applications:

1. Introduction to ASP.NET Core.
2. Creating basic APIs and understanding HTTP methods.
3. Implementing authentication and authorization.
4. Basics of Model-View-Controller (MVC) in web applications.

10. Project Deployment and Documentation:

1. Overview of deployment models: local, server, and cloud.
2. Understanding the importance of documentation.
3. Best practices in writing and maintaining good documentation.
4. Continuous integration/continuous deployment (CI/CD) basics.

11. Final Project:

1. Participants will apply what they've learned by creating a small project.
2. The project encourages innovative problem solving and the application of various concepts covered during the training.
3. Trainers will provide guidance and feedback, mimicking real-world project oversight.

Each of these topics will be taught through a blend of lectures, hands-on exercises, group discussions, and individual assessments to ensure an interactive and comprehensive learning journey. Participants will also have the opportunity to engage in Q&A sessions at regular intervals, allowing for a dynamic exchange of ideas and solutions with experienced professionals. This approach ensures that learning is not just theoretical but grounded in practical experience and industry realities.

© 2026 Faysal Aziz · Kuala Lumpur