Course description

This practical five-day course is aimed at developers who rapidly need to gain the skills to develop robust commercial applications, using the latest features of the .NET framework. Database-connected web and rich client applications are assembled, using Entity Framework Core to implement the data layer with ASP.NET Core MVC and WPF for the presentation layers. Other topics include the collections classes, LINQ, test driven development, dependency injection, concurrent programming and distributed revision control.

Venue and fees

Course hours are 9:00 to 5:00 at air-conditioned offices in Reading. There are a maximum of eight delegates and each has the use of a computer. As well as the above scheduled public courses, we can deliver customised on-site training for a fee of £4800+VAT for 5 days. This includes all expenses and the use of laptops, if required.

What you will learn
- Fundamentals of object-oriented programming
- Test Driven Development
- Collections and LINQ
- Object Mapping with Entity Framework Core
- Concurrent and Asynchronous Programming
- ASP.NET Core MVC and Web API 2
- Windows Presentation Foundation
- Design Principes and Patterns
- Distributed Revision Control

Prerequisites

Developers who have previously used a programming language; not necessarily an object-oriented language

Course instructor

Originally a Civil Engineer with a BSc from London University the course instructor, Simon Dineen, has 15 years' experience in .NET training and development.

Contact

To discuss training options or to make a booking, please telephone 0118 966 4994 or email mailbox@javaconsult.co.uk

Course outline

Introduction to the .NET framework
- The .NET Standard specification
- The .NET Core and .NET Framework implementations
- Common Intermediate Language
- The Common Type System
- The Common Language Specification

Core language features
- Value types, reference types, conversion and casting
- Operators and decision structures
- Repetition including for, while and recursion

Classes and objects
- UML class and object diagrams
- Fields, properties, auto-implemented properties and static methods
- Constructors and method overloading
- Viewing compiled code with the intermediate language disassembler

Inheritance
- Reasons for creating derived classes
- Access modifiers
- Overriding methods and calling base class constructors
- Object reference conversion and polymorphism
- Methods in the Object class, reference equality and value equality

Unit Testing
- Overview of Test-driven development (TDD)
- Writing unit tests with xUnit
- Parameterised tests
- Writing interactions tests with Moq

Arrays and Collections
- Defining and implementing interfaces
- Classes and interfaces in the System.Collections.Generic namespace
- Extension methods, delegates and lambda expressions
- Interrogating collections with LINQ Query syntax and method syntax

File Handling and Exceptions
- Reading and writing text to a file
- Serializing objects to JSON
- try catch and finally blocks
- using blocks and the autocloseable interface
- throwing exceptions

SQL
- Classes and interfaces in System.Data.SqlClient
- Transact-SQL data types, data definition and data manipulation
- Transactions and pessimistic concurrency
- Using SQL joins to combine rows from two tables

Entity Framework Core
- Overview of object-relational mapping
- Using DbContext, DbSet and Entity State
- Code-first migrations and data annotations
- RowVersion columns and optimistic concurrency
- Repository and Service layers

ASP.NET Core MVC
- Controllers, actions and routing
- Defining views using the Razor view engine
- Scaffolding, forms, Tag helpers and model binding
- Validation with jQuery
- Dependency injection
- Authentication with ASP.NET Core Identity
- Unit testing the Controller

Web API 2
- Overview of web services, HTTP methods, REST and JSON
- HTTP status codes and ApiController methods
- Model binding and handling exceptions
- Attribute Routing and constraints
- Testing web services with the Postman HTTP client
- Writing integration tests for the web service

WPF
- Overview of Windows Presentation Foundation
- Controls, properties and events
- Layouts, styles and resources
- Tasks, asynchronous methods and the await keyword
- Call the Web API built previously

WPF MVVM

- Benefits of the Model-View-ViewModel pattern
- Data binding and change notification events
- Dependency injection with the unity container
- Unit tests with xUnit and Moq

Design Principles and Patterns

- Object-oriented design principles, including SOLID
- Creational design patterns including Factory, Abstract Factory and Singleton
- Structural patterns including Adapter, Decorator and Facade
- Behavioural patterns including Observer, Strategy and Template Method