

Programming in C# (20483)

- **Formato do curso:** Presencial
- **Preço:** 1600€
- **Nível:** Intermédio
- **Duração:** 35 horas

Este curso permite aos formandos adquirir as competências para desenvolver aplicações Windows com a linguagem de programação C#. Durante os cinco dias do curso, os formandos fazem uma revisão das bases da linguagem C# como também consolidarão os conhecimentos através do desenvolvimento de uma aplicação que incorpora várias funcionalidades da Framework .NET 4.5.

Destinatários

- Programadores com experiência numa das seguintes linguagens:
 - C
 - C++
 - JavaScript
 - Objective-C
 - Microsoft Visual Basic
 - Java
- Quem esteja interessado em fazer o exame 70-483

Pré-requisitos

- Conhecimentos base da linguagem de programação C#, tais como:
 - Nomear, declarar, inicializar e atribuir valores a variáveis;
 - Operadores aritméticos, relacionais e lógicos;
 - Sintaxe base do C# e reconhecer os erros da mesma com o IDE do Visual Studio;
 - Estruturas de decisão (If);
 - Estruturas de repetição (For);
 - Utilizar o IDE do Visual Studio para identificar erros lógicos;
 - Criar métodos com parâmetros e valores de retorno;
 - Criar um interface gráfico simples através da toolbox do Visual Studio;
 - Conectar-se a uma base de dados SQL Server e as operações base de CRUD;
 - Ordenar dados num ciclo;
 - Reconhecer as classes e os métodos utilizados numa aplicação.
- Conhecimentos base em programação orientada a objetos.

Objectivos

- Descrever a sintaxe base e funcionalidades do C#-
- Criar e invocar métodos, agarrar e tratar exceções e descrever a monitorização dos requisitos de aplicações de larga escala.
- Implementar a estrutura básica e os elementos essenciais de uma aplicação de desktop típica.
- Criar classes, definir e implementar interfaces e criar e utilizar coleções genéricas.
- Utilizar herança para uma hierarquia de classes, expandir as classes da Framework .NET, criar classes e métodos genéricos.
- Ler e escrever dados através de ficheiros e streams e a serializar e a deserializar dados em formatos diferentes.
- Criar e utilizar modelos de dados Entity para aceder a base de dados e utilizar LINQ para pesquisar e modificar dados.
- Utilização de tipos no namespace System.Net e WCF Data Services para aceder e pesquisar dados remotos.
- Construir um interface gráfico através de XAML.
- Melhorar o processamento e o tempo de resposta das aplicações através de tasks e operações assíncronas.
- Integrar unmanage libraries e componentes dynamic numa aplicação C#.
- Examinar os metadados dos tipos através de reflection, criar e utilizar atributos, gerar código em runtime e gerir as versões das assemblies.
- Encriptar e desencriptar através de encriptação simétrica e assimétrica.

Metodologia

Pode assistir a este curso de forma:

- Presencial
- [Live Training](#)
- [MOC On-demand](#)

Programa

Review of C# Syntax

This module reviews the core syntax and features of the C# programming language. It also provides an introduction to the Visual Studio 2012 debugger.

- Lessons Overview of Writing Applications using C# Datatypes, Operators, and Expressions C# Programming Language Constructs
- Lab : Developing the Class Enrolment Application Implementing Edit Functionality for the Students List Implementing Insert Functionality for the Students List Implementing Delete Functionality for the Students

List Displaying the Student Age

Creating Methods, Handling Exceptions, and Monitoring Applications

This module explains how to create and call methods, catch and handle exceptions. This module also describes the monitoring requirements of large-scale applications.

- Lessons Creating and Invoking Methods Creating Overloaded Methods and Using Optional and Output Parameters Handling Exceptions Monitoring Applications
- Lab : Extending the Class Enrolment Application Functionality Refactoring the Enrolment Code Validating Student Information Saving Changes to the Class List

Developing the Code for a Graphical Application

This module describes how to implement the basic structure and essential elements of a typical desktop application, including using structures and enumerations, collections, and events.

- Lessons Implementing Structs and Enums Organizing Data into Collections Handling Events
- Lab : Writing the Code for the Grades Prototype Application Adding Navigation Logic to the Application Creating Data Types to Store User and Grade Information Displaying User and Grade Information

Creating Classes and Implementing Type-safe Collections

This module explains how to create classes, define and implement interfaces, and create and use generic collections. This module also describes the differences between value types and reference types in C#.

- Lessons Creating Classes Defining and Implementing Interfaces Implementing Type-safe Collections
- Lab : Adding Data Validation and Type-safety to the Grades Application Implementing the Teacher, Student, and Grade Types as Classes Adding Data Validation to the Grade Class Displaying Students in Name Order Enabling Teachers to Modify Class and Grade Data

Creating a Class Hierarchy by Using Inheritance

This module explains how to use inheritance to create a class hierarchy and extend a .NET Framework class. This module also describes how to create generic classes and define extension methods.

- Lessons Creating Class Hierarchies Extending .NET Framework Classes Creating Generic Types
- Lab : Refactoring Common Functionality into the User Class Creating and Inheriting from the User Base Class Implementing Password Complexity by Using an Abstract Method Creating the ClassFullException Class

Reading and Writing Local Data

This module explains how to read and write data by using file input/output (I/O) and streams, and how to serialize and deserialize data in different formats.

- Lessons Reading and Writing Files Serializing and Deserializing Data Performing I/O Using Streams
- Lab : Generating the Grades Report Serializing the Data for the Grades Report as XML Previewing the Grades Report Persisting the Serialized Grades Data to a File

Accessing a Database

This module explains how to create and use an entity data model for accessing a database, and how to use LINQ to query and update data.

- Lessons Creating and Using Entity Data Models Querying Data by Using LINQ Updating Data by Using LINQ
- Lab : Retrieving and Modifying Grade Data Creating an Entity Model from the The School of Fine Arts Database Updating Student and Grade Data Using the Entity Framework Extending the Entity Model to Validate Data

Accessing Remote Data

This module explains how to use the types in the System.Net namespace, and WCF Data Services, to query and modify remote data.

- Lessons Accessing Data Across the Web Accessing Data in the Cloud
- Lab : Retrieving and Modifying Grade Data in the Cloud Creating a WCF Data Service for the SchoolGrades Database Integrating the WCF Data Service into the Application Retrieving Student Photographs Over the Web (if time permits)

Designing the User Interface for a Graphical Application

This module explains how to build and style a graphical user interface by using XAML. This module also describes how to display data in a user interface by using data binding.

- Lessons Using XAML to Design a User Interface Binding Controls to Data Styling a User Interface
- Lab : Customizing Student Photographs and Styling the Application Customizing the Appearance of Student Photographs Styling the Logon View Animating the StudentPhoto Control (If Time Permits)

Improving Application Performance and Responsiveness

This module explains how to improve the throughput and response time of applications by using tasks and asynchronous operations.

- Lessons Implementing Multitasking by using Tasks and Lambda Expressions Performing Operations Asynchronously Synchronizing Concurrent Access to Data
- Lab : Improving the Responsiveness and Performance of the Application Ensuring that the User Interface Remains Responsive When Retrieving Data for Teachers Providing Visual Feedback During Long-Running Operations

Integrating with Unmanaged Code

This module explains how to integrate unmanaged libraries and dynamic components into a C# application. This module also describes how to control the lifetime of unmanaged resources.

- Lessons Creating and Using Dynamic Objects Managing the Lifetime of Objects and Controlling Unmanaged Resources
- Lab : Upgrading the Grades Report Generating the Grades Report by Using Microsoft Office Word Controlling the Lifetime of Word Objects by Implementing the Dispose Pattern

Creating Reusable Types and Assemblies

This module explains how to examine the metadata of types by using reflection, create and use custom attributes, generate managed code at runtime, and manage different versions of assemblies.

- Lessons Examining Object Metadata Creating and Using Custom Attributes Generating Managed Code Versioning, Signing and Deploying Assemblies
- Lab : Specifying the Data to Include in the Grades Report Creating the IncludeInReport Attribute Generating the Report Storing the Grades. Utilities Assembly Centrally

Encrypting and Decrypting Data

This module explains how to encrypt and decrypt data by using symmetric and asymmetric encryption.

- Lessons Implementing Symmetric Encryption Implementing Asymmetric Encryption
- Lab : Encrypting and Decrypting Grades Reports Encrypting the Grades Report Decrypting the Grades Report