

Corso

Programming in C#



Durata: 4-5 giornate

A chi è rivolto:

Ai programmatori con esperienza in ambito C, C++, JavaScript, Visual Basic, Java

Scopo del corso:

Il corso è rivolto a chi vuole acquisire le competenze necessarie alla creazione di applicazioni Windows con C#. Agli allievi vengono presentati la struttura della programmazione C#, la sintassi e i dettagli implementativi.

Durante i laboratori vengono presentate diverse funzionalità e caratteristiche del .NET Framework 4.5. Il corso introduce diverse tecniche e tecnologie utilizzate dalle più recenti applicazioni desktop e enterprise, quali:

- creazione di nuovi tipi di dato;
- gestione degli eventi;
- programmazione della interfaccia utente;
- accesso ai database;
- utilizzo di dati remoti;
- esecuzione di operazioni asincrone;
- creazione di attributi custom;
- criptazione e decriptazione di dati.

Gli obiettivi sono i seguenti:

- implementare struttura e elementi di una classica applicazione desktop;
- creare classi, definire l'interfaccia, creare e usare collection;

- utilizzare l'ereditarietà per creare una gerarchia di classi;
- leggere e scrivere dati tramite file I/O e stream;
- utilizzare entity data model e LINQ;
- interrogare dati remoti tramite WCF Data Services e System.Net namespace;
- creare una interfaccia utente tramite XAML;
- integrare librerie e componenti dinamici.

Prerequisiti:

Occorre che gli allievi abbiano già una esperienza di base con C# per quanto riguarda:

- dichiarazione, inizializzazione e assegnazione di una variabile;
- uso di operatori aritmetici, operatori relazionali e operatori logici;
- scrittura di semplici statement di programmazione;
- creazione di statement IF;
- creazione di loop;
- creazione di funzioni;
- creazione di una semplice interfaccia utente;
- creazione di una connessione a un database SQL Server;
- individuazione delle classi e dei metodi utilizzati in un programma.

Contenuti:

Module 1: 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.

Module 2: 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.

Module 3: 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.

Module 4: 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#.

Module 5: 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.

Module 6: 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.

Module 7: 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.

Module 8: 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.

Module 9: 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.



Scheda corso "Programming in C#"
Per informazioni www.greensistemi.it - info@greensistemi.it
Ogni corso è personalizzabile in base alle vostre esigenze.

Module 10: Improving Application Performance and Responsiveness

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

Module 11: 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.

Module 12: 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.

Module 13: Encrypting and Decrypting Data

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

Certificazioni

Il corso è propedeutico per i seguenti esami:

- 70-483 - Programming in C#