Pipeline

Pipeline is a Leading Learning Partners Association Member

VOUCHER ESAME C# CODING SPECIALIST (CCS) - KNOWLEDGE PILLARS

VOUCHER ESAME

Durata	Prezzo	Orari	Calendario
	150,00€ + IVA		

La **certificazione C# Coding Specialist** di Knowledge Pillars è una certificazione di settore riconosciuta a livello mondiale sulla programmazione C#, che permette a programmatori, sviluppatori di software o ingegneri, sviluppatori di giochi e professionisti IT la possibilità di valutare le loro conoscenze e ottenere credenziali relativamente alle loro abilità di programmazione. C# è molto versatile, e può essere usato per creare i progetti più diversi, incluse le applicazioni mobili, servizi basati su cloud, software aziendali e giochi.

I candidati ideali a questo esame sono sviluppatori con almeno un anno di esperienza nella programmazione utilizzando C#. I candidati dovrebbero avere una comprensione approfondita di quanto segue:

- Gestione del flusso del programma e degli eventi
- Programmazione asincrona e threading
- Convalida dei dati e lavoro con le data collection, incluso LINQ
- Gestione degli errori e delle eccezioni
- Lavorare con gli array e le collection
- Operare con variabili, operatori ed espressioni
- Lavorare con classi e metodi
- Dichiarazioni di decisione e iterazione

Puoi preparare questo esame frequentando il corso MOC 20483 Programming in C#.

Acquisto dell'esame

In questa pagina è possibile acquistare il voucher (codice elettronico) per procedere con l'iscrizione. Una volta ricevuto il voucher via email, l'iscrizione all'esame C# Coding Specialist (CCS) va fatta sul portale Knowledge Pillars.

Il prezzo indicato è al netto di IVA.

Caratteristiche dell'esame per conseguire la certificazione C# Coding Specialist

Numero di domande: 35
Tempo massimo: 45 minuti

Punteggio minimo: 75%
Esame valido per: 2 anni
Lingua dell'esame: inglese

I risultati dell'esame vengono inviati ai candidati entro 72 ore.

Modalità di svolgimento dell'esame

La peculiarità della Certificazione CCS è la modalità di valutazione, che consiste in un ambiente di codifica reale dal vivo, offre ai candidati la possibilità di scrivere codice corretto ed eseguire domande basate su compiti. Questa tecnologia è chiamata LITA (Live in the Application) e Knowledge Pillars è una delle poche organizzazioni in grado di fornire questa soluzione di valutazione molto avanzata.
Gli studenti dovranno rispondere a tutte le domande entro il tempo stabilito.
Sono liberi di ignorare tutte le domande che vogliono. Avranno la possibilità di rispondere alle domande ignorate alla fine del test (entro i 45 minuti).
Tutte le domande senza risposta saranno segnate come errate. Quando il test finisce, il risultato dell'esame viene inviato al server di KP per l'archiviazione. Se gli esaminandi inseriscono nuovamente la loro credenziale, potranno vedere i risultati per ogni loro voucher.

ESAME: contenuti tecnici

1. Manage Program Flow

- **Implement multithreading and asynchronous processing**

Use the Task Parallel library, including theParallel.For method, PLINQ, Tasks; create continuation tasks; spawn threads by using ThreadPool; unblock the UI; use async and await keywords; manage data by using concurrent collections

- **Manage multithreading**

Synchronize resources; implement locking; cancel a long-running task; implement thread-safe methods to handle race conditions

- **Implement program flow**

Iterate across collection and array items; program decisions by using switch statements, if/then, and operators; evaluate expressions

- **Create and implement events and callbacks**

Create event handlers; subscribe to and unsubscribe from events; use built-in delegate types to create events; create delegates; lambda expressions; anonymous methods

- **Implement exception handling**

Handle exception types, including SQL exceptions, network exceptions, communication exceptions, network timeout exceptions; use catch statements; use base class of an exception; implement try-catch-finally blocks; throw exceptions; rethrow an exception; create custom exceptions; handle inner exceptions; handle aggregate exceptions

2. Create and Use Types

- **Create types**

Create value types, including structs and enum; create reference types, generic types, constructors, static variables, methods, classes, extension methods; create optional and named parameters; create indexed properties; create overloaded and overridden methods

- **Consume types**

Box or unbox to convert between value types; cast types; convert types; handle dynamic types; ensure interoperability with code that accesses COM APIs

- **Enforce encapsulation**

Enforce encapsulation by using properties; enforce encapsulation by using accessors, including public, private, protected, and internal; enforce encapsulation by using explicit interface implementation

- **Create and implement a class hierarchy**

Design and implement an interface; inherit from a base class; create and implement classes based on the IComparable, IEnumerable, IDisposable, and IUnknown interfaces

- **Find, execute, and create types at runtime by using reflection**

Create and apply attributes; read attributes; generate code at runtime by using CodeDom and Lambda expressions; use types from the System.Reflection namespace, including Assembly, PropertyInfo, MethodInfo, Type

- **Manage the object life cycle**

Manage unmanaged resources; implement IDisposable, including interaction with finalization; manage IDisposable by using the Using statement; manage finalization and garbage collection

- **Manipulate strings**

Manipulate strings by using the StringBuilder, StringWriter, and StringReader classes; search strings; enumerate string methods; format strings; use string interpolation

3. Debug Applications and Implement Security

- **Validate application input**

Validate JSON data; choose the appropriate data collection type; manage data integrity; evaluate a regular expression to validate the input format; use built-in functions to validate data type and content

- **Perform symmetric and asymmetric encryption**

Choose an appropriate encryption algorithm; manage and create certificates; implement key management; implement the System.Security namespace; hashing data; encrypt streams

- **Manage assemblies**

Version assemblies; sign assemblies using strong names; implement side-by-side hosting; put an assembly in the global assembly cache; create a WinMD assembly

- **Debug an application**

Create and manage preprocessor directives; choose an appropriate build type; manage program database files (debug symbols)

- **Implement diagnostics in an application**

Implement logging and tracing; profiling applications; create and monitor performance counters; write to the event log

4. Implement Data Access

- **Perform I/O operations**

Read and write files and streams; read and write from the network by using classes in the System.Net namespace; implement asynchronous I/O operations

- **Consume data**

Retrieve data from a database; update data in a database; consume JSON and XML data; retrieve data by using web services

- **Query and manipulate data and objects by using LINQ**

Query data by using operators, including projection, join, group, take, skip, aggregate; create method-based LINQ queries; query data by using query comprehension syntax; select data by using anonymous types; force execution of a query; read, filter, create, and modify data structures by using LINQ to XML

- **Serialize and deserialize data**

Serialize and deserialize data by using binary serialization, custom serialization, XML Serializer, JSON Serializer, and Data Contract Serializer

- **Store data in and retrieve data from collections**

Store and retrieve data by using dictionaries, arrays, lists, sets, and queues; choose a collection type; initialize a collection; add and remove items from a collection; use typed vs. non-typed collections; implement custom collections; implement collection interfaces

Pipeline

Pipeline is a Leading Learning Partners Association Member

Vuoi organizzare un corso personalizzato?

Chiamaci: 02/6074791 Scrivici: formazione@pipeline.it

MAIN PARTNERS

Microsoft Solutions Partner

CompTIA Authorized Partner DELIVERY PARTNER

ITIL