

20483: Programming in C#

Duración: **25 Horas**

INTRODUCCION:

Esta formación enseña a los asistentes las habilidades de programación que se requieren para que los desarrolladores creen aplicaciones de Windows utilizando el lenguaje Visual C#. Durante la formación los asistentes revisan los conceptos básicos de la estructura del programa Visual C#, la sintaxis del lenguaje y los detalles de implementación, y luego consolidan sus conocimientos durante la semana a medida que crean una aplicación que incorpora varias características de .NET Framework 4.7.

DIRIGIDO A:

Desarrolladores con experiencia que ya tienen experiencia en programación en C, C ++, JavaScript, Objective-C, Microsoft Visual Basic o Java y comprenden los conceptos de programación orientada a objetos.

OBJETIVOS:

Al finalizar la formación los asistentes los asistentes serán capaces de:

- 🔗 Describir la sintaxis central y las características de Visual C#.
- 🔗 Crear métodos, manejar excepciones y describir los requisitos de supervisión de aplicaciones a gran escala.
- 🔗 Implementar la estructura básica y los elementos esenciales de una aplicación de escritorio típica.
- 🔗 Crear clases, defina e implemente interfaces, y crear y usar colecciones genéricas.
- 🔗 Usar la herencia para crear una jerarquía de clases y extender una clase de .NET Framework.
- 🔗 Leer y escribir datos utilizando la entrada/salida de archivos y flujos, y serializar y deserializar datos en diferentes formatos.
- 🔗 Crear y usar un modelo de datos de entidad para acceder a una base de datos y usar LINQ para consultar datos.
- 🔗 Acceder y consultar datos remotos utilizando los tipos en el espacio de nombres System.Net y WCF Data Services.
- 🔗 Crear una interfaz gráfica de usuario utilizando XAML.
- 🔗 Mejorar el rendimiento y el tiempo de respuesta de las aplicaciones mediante el uso de tareas y operaciones asincrónicas.
- 🔗 Integrar bibliotecas no administradas y componentes dinámicos en una aplicación de Visual C#.
- 🔗 Examinar los metadatos de los tipos mediante la reflexión, cree y usar atributos personalizados, generar código en tiempo de ejecución y administrar versiones de ensamblaje.
- 🔗 Cifrar y descifrar datos mediante el cifrado simétrico y asimétrico.

EXAMEN:










Esta formación es la recomendada por Microsoft para la preparación del examen:

70-483: Programming in C#

Este examen se retirará el próximo día 31 de enero de 2021 junto con el resto de exámenes asociados a las certificaciones MCSA, MCSD y MCSE.




REQUISITOS:

Los desarrolladores que asisten a esta formación ya deberían haber adquirido una experiencia limitada usando C# para completar tareas básicas de programación. Más específicamente, los estudiantes deben tener experiencia práctica en el uso de C# que demuestre su comprensión de lo siguiente:





-  Cómo nombrar, declarar, inicializar y asignar valores a variables dentro de una aplicación.
-  Cómo usar: operadores aritméticos para realizar cálculos aritméticos con una o más variables; operadores relacionales para probar la relación entre dos variables o expresiones; operadores lógicos para combinar expresiones que contienen operadores relacionales.
-  Cómo crear la sintaxis de código para declaraciones de programación simples usando palabras clave del lenguaje C# y reconocer errores de sintaxis usando el IDE de Visual Studio.
-  Cómo crear una estructura de ramificación simple usando una declaración IF.
-  Cómo crear una estructura de bucle simple usando una instrucción para iterar a través de una matriz de datos.
-  Cómo usar el IDE de Visual Studio para localizar errores lógicos simples.
-  Cómo crear una función que acepte argumentos (parámetros y devuelva un valor de un tipo especificado).
-  Cómo diseñar y construir una interfaz de usuario simple utilizando controles estándar de la caja de herramientas de Visual Studio.
-  Cómo conectarse a una base de datos de SQL Server y los conceptos básicos sobre cómo recuperar y almacenar datos.

CONTENIDO:

Módulo 1: Revisión de la sintaxis de Visual C#.

-  Descripción general de la aplicación de escritura mediante Visual C#.
-  Tipos de datos, operadores y expresiones.
-  Construcciones del lenguaje de programación Visual C#.

Módulo 2: Creación de métodos, manejo de excepciones y aplicaciones de supervisión.

-  Crear e invocar métodos.
-  Creación de métodos sobrecargados y uso de parámetros opcionales y de salida.
-  Manejo de excepciones.
-  Supervisión de aplicaciones.

Módulo 3: Tipos básicos y construcciones de Visual C#.

- 🔗 Implementación de estructuras y enumeraciones.
- 🔗 Organización de datos en colecciones.
- 🔗 Manejo de eventos.

Módulo 4: Crear clases e implementar colecciones de tipos seguros.

- 🔗 Crear clases.
- 🔗 Definición e implementación de interfaces.
- 🔗 Implementación de colecciones de tipo seguro.

Módulo 5: Crear una jerarquía de clases mediante el uso de herencia.

- 🔗 Crear jerarquías de clases.
- 🔗 Extender las clases de .NET Framework.

Módulo 6: Leer y escribir datos locales.

- 🔗 Lectura y escritura de archivos.
- 🔗 Serialización y deserialización de datos.
- 🔗 Realización de E/S mediante el uso de secuencias.

Módulo 7: Acceder a base de datos.

- 🔗 Creación y uso de modelos de datos de entidad.
- 🔗 Consulta de datos mediante LINQ.

Módulo 8: Acceder a datos remotos.

- 🔗 Acceso a datos a través de la web.
- 🔗 Acceso a datos mediante el uso de servicios conectados de Oda.

Módulo 9: Diseño de la interfaz de usuario para una aplicación gráfica.

- 🔗 Uso de XAML para diseñar una interfaz de usuario.
- 🔗 Controles vinculantes a datos.





Módulo 10: Mejorar el rendimiento de la aplicación y la capacidad de respuesta.

- 🔗 Implementación de tareas múltiples.
- 🔗 Realización de operaciones asincrónicas.
- 🔗 Sincronización de acceso concurrente a datos.



Módulo 11: Integración con código no administrado.

- 🔗 Crear y usar objetos dinámicos.
- 🔗 Administrar la vida útil de los objetos y controlar los recursos no administrados.

Módulo 12: Crear tipos y ensamblados reutilizables.

-  Examen de metadatos de objetos.
-  Crear y usar atributos personalizados.
-  Generar código administrado.
-  Versiones, firma y despliegue de ensamblados.

Módulo 13: Cifrar y descifrar datos.

-  Implementación de cifrado simétrico.
-  Implementación de cifrado asimétrico.