

Experto en C++, avances en el lenguaje (de C++ 11 a 20)

Curso práctico de 3 días - 21h

Ref.: VEC - Precio 2025: 1 330€ sin IVA

Este curso le permitirá asimilar las nuevas características introducidas por los nuevos estándares C++. Abarcará las posibilidades de la programación funcional, dominará la gestión de memoria con punteros inteligentes y explotará las demás novedades de la biblioteca estándar de C++.

OBJETIVOS PEDAGÓGICOS

Al término de la formación, el alumno podrá:

Comprender las nuevas funciones y mejoras del lenguaje C++

Uso de expresiones lambda

Mejora del rendimiento con la semántica de desplazamiento

Gestión de la asignación y destrucción de objetos con punteros inteligentes

Desarrollar una aplicación multihilo

Los ejercicios prácticos de programación le ayudarán a asimilar los distintos conceptos tratados.

PROGRAMA

última actualización: 08/2024

1) La llegada de C++11

- Las distintas normas C++98, C++03, C++11, C++14, C++17, C++20.
 - Novedades de C++11 y objetivos de esta norma. El futuro de Boost y STL.
 - La cuestión de la compatibilidad de los códigos antiguos.
 - La disponibilidad de herramientas de desarrollo (compiladores, depuradores, IDE, etc.).
- Trabajo práctico* : Verificación de la herramienta utilizando el código C++11 proporcionado.

2) Mejoras lingüísticas

- Enums fuertemente tipados.
 - La palabra clave auto y decltype para simplificar la escritura.
 - El bucle basado en intervalos.
 - Coroutines C++20.
 - constexpr para la evaluación en tiempo de compilación.
 - Módulos C++20.
 - Plantillas variadas, conceptos de plantillas C++20...
- Trabajo práctico* : Aplicación de mejoras.

3) Cambios en las clases

- Delegación de constructores y restricciones de herencia.
 - La nueva semántica del desplazamiento y el constructor de desplazamiento.
 - Las directivas override, final =delete, =default.
 - Noción de módulo C++20
- Trabajo práctico* : Creación de clases.

PARTICIPANTES

Desarrolladores de C++ que deseen conocer las nuevas características definidas por la norma C++11.

REQUISITOS PREVIOS

Buen conocimiento del lenguaje C++. Se recomienda experiencia práctica en programación con este lenguaje.

COMPETENCIAS DEL FORMADOR

Los expertos que imparten la formación son especialistas en las materias tratadas. Han sido validados por nuestros equipos pedagógicos, tanto en el plano de los conocimientos profesionales como en el de la pedagogía, para cada curso que imparten. Cuentan al menos con entre cinco y diez años de experiencia en su área y ocupan o han ocupado puestos de responsabilidad en empresas.

MODALIDADES DE EVALUACIÓN

El formador evalúa los progresos pedagógicos del participante a lo largo de toda la formación mediante preguntas de opción múltiple, escenificaciones de situaciones, trabajos prácticos, etc. El participante también completará una prueba de posicionamiento previo y posterior para validar las competencias adquiridas.

MEDIOS PEDAGÓGICOS Y TÉCNICOS

- Los medios pedagógicos y los métodos de enseñanza utilizados son principalmente: ayudas audiovisuales, documentación y soporte de cursos, ejercicios prácticos de aplicación y ejercicios corregidos para los cursos prácticos, estudios de casos o presentación de casos reales para los seminarios de formación.
- Al final de cada curso o seminario, ORSYS facilita a los participantes un cuestionario de evaluación del curso que analizarán luego nuestros equipos pedagógicos.
- Al final de la formación se entrega una hoja de presencia por cada media jornada de presencia, así como un certificado de fin de formación si el alumno ha asistido a la totalidad de la sesión.

MODALIDADES Y PLAZOS DE ACCESO

La inscripción debe estar finalizada 24 horas antes del inicio de la formación.

ACCESIBILIDAD DE LAS PERSONAS CON DISCAPACIDAD

¿Tiene alguna necesidad específica de accesibilidad? Póngase en contacto con la Sra. FOSSE, interlocutora sobre discapacidad, en la siguiente dirección psh-accueil@orsys.fr para estudiar de la mejor forma posible su solicitud y su viabilidad.

4) Utilización de hilos

- Declarar y ejecutar un hilo. Esperar el final de la ejecución con join().
- Gestión de los datos locales de un hilo.
- Recuperación de un resultado con future<> y async().

Trabajo práctico : Trabajo práctico

5) Contenedores unordered_set y unordered_map basados en hash.

- Rangos con std::view para la evaluación sobre la marcha de C++20.
- Gestión del tiempo con el espacio de nombres chrono.
- Formateo de cadenas con C++20.

Trabajo práctico : Implementación de nuevas funciones.

6) Programación funcional con expresiones lambda

- Declaración, tipificación, aplicación y uso.

Trabajo práctico : Gestión del cierre, con captura por valor o por referencia de variables vinculadas al contexto.

FECHAS

Contacto