# *Programación de objetos en C*

## *Curso práctico de 5 días - 35h*
## *Ref.: C++ - Precio 2024: 2 070€ sin IVA*

Este curso intensivo persigue dos objetivos: introducir a los participantes en los métodos y reflejos de la programación orientada a objetos y proporcionarles un dominio operativo completo del lenguaje C++. Es el fruto de muchos años de experiencia en el desarrollo en C++ y se articula en torno a un método pedagógico riguroso basado en un gran número de ejercicios prácticos progresivos. Durante estos 5 días, el formador presentará las evoluciones de las normas desde C++98 hasta C++20.

### OBJETIVOS PEDAGÓGICOS

Al término de la formación, el alumno podrá:

Dominar la sintaxis del lenguaje C

Aplicar los conceptos del diseño orientado a objetos

Utilizar las herramientas de desarrollo asociadas al lenguaje C++

Dominar las principales novedades de la norma C++ 11

### MÉTODOS PEDAGÓGICOS

Todos los ejercicios incluyen una fase de análisis/diseño seguida de una fase de programación.

Estaciones de trabajo equipadas con Visual C++ (Windows) y gcc (Unix). Los ejercicios prácticos se han diseñado para ilustrar todos los elementos del lenguaje y aplicar sistemáticamente los conceptos del diseño orientado a objetos.

### PARTICIPANTES

Desarrolladores, ingenieros y gestores de proyectos implicados en el desarrollo.

### REQUISITOS PREVIOS

Buen conocimiento de un lenguaje de programación como C, Java, Python, C#, VB.NET o PHP.

### 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 cursillos prácticos, estudios de casos o presentación de casos reales para los seminarios de formación.
• Al final de cada cursillo 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.

## PROGRAMA

última actualización: 09/2023

### 1) Sintaxis C++ (diferencias entre C y C++)

- Datos: definición, inicialización, tipos de datos.
- Expresiones: noción de referencia, mecanismos de reparto.
- Operadores (: :, nuevo, suprimir).
- Funciones (paso de parámetros y valores de retorno por referencia, valores por defecto, inlining, sobrecarga).
- Uso de código C en un programa C++.
- Referencias (argumentos y valores de retorno).
- Tipos constantes.
- Espacios de nombres.
- Escritura "automática" con la palabra clave auto (C++ 11).

*Trabajo práctico : Familiarizarse con el entorno de desarrollo y programar un programa sencillo.*

### 2) Enfoque orientado a objetos

- Los principios generales de las técnicas Objet.
- C++ y programación orientada a objetos.
- Introducción a las metodologías orientadas a objetos.
- Introducción a los modelos y a la notación UML (modelo estático, modelo dinámico, modelo de cooperación, escenario).

*Trabajo práctico : Aplicación de los conceptos a un estudio de caso, que será uno de los temas principales de los ejercicios siguientes.*

### 3) Clases y objetos C
- Sintaxis: campos, métodos, constructores.
- Control de acceso.
- Autorreferencia.
- Campos y métodos estáticos.
- Las funciones.
- Métodos y clases amigas.
- Creación dinámica de matrices de objetos.
- Aspectos metodológicos: diseño de la clase.
- Copiar y mover constructores (C++11).
- Delegación de constructores (C++ 11).
- Introducción a la gestión de la memoria (pila, montón, recolector de basura, etc.).

*Trabajo práctico : Programación del caso práctico. Diseñar y construir una jerarquía de clases e interfaces.*

### 4) Derivación y herencia
- Principio de derivación.
- Sintaxis: definición de clases derivadas, constructores.
- Control de acceso.
- Implementación del polimorfismo: funciones virtuales.
- Reutilización del código: clases abstractas.
- Interfaces.
- Derivación múltiple.
- Aspectos semánticos y metodológicos: factorizar el código.

*Trabajo práctico : La aplicación del polimorfismo en el estudio de casos.*

### 5) Excepciones
- Sintaxis: bloques try, generación de excepciones.
- Aspectos metodológicos: construcción de una jerarquía de excepciones, utilización de excepciones.

*Trabajo práctico : La introducción de excepciones en el estudio de casos.*

### 6) Sobrecarga del operador
- Principio de sobrecarga.
- Sobrecarga del operador binario.
- Sobrecarga especial: el operador de índice, función y conversión.
- Sobrecarga de operadores de gestión de memoria.
- Sobrecarga de los operadores '<<' et '>>'.

*Trabajo práctico : Sobrecarga de algunos operadores sencillos.*

### 7) Los modelos
- Modelo de clase. Principios generales y mecanismos. Sobrecarga del modelo y redefinición de métodos.
- Modelo de función. Principios generales y mecanismos. Sobrecarga de modelos.
- Modelos y sobrecarga del operador.
- Modelos y mecanismos de derivación.
- Las mejoras que ofrece C++ 11.

*Trabajo práctico : Ejercicios sobre modelos.*

### 8) Resumen de E/S y STL
- E/S.
- El principio de los flujos y la jerarquía de las clases de entrada/salida.
- Descripción de algunas clases de entrada/salida.
- Visión general de STL.
- Objetivos y principios.
- Descripciones de algunos modelos y clases.

- Contenedores, iteradores, bucles basados en intervalos (C++ 11).

### 9) Conclusión

- Ciclo de vida del software: pruebas, integración, método de lanzamiento a producción.
- Interacción con otros entornos.
- Análisis crítico de C++.
- Evolución de C++.

## FECHAS

Contacto