

Programación en C

Curso práctico de 5 días - 35h

Ref.: LGC - Precio 2026: 2 150€ sin IVA

Este curso intensivo le permitirá adquirir un conocimiento verdaderamente operativo del lenguaje. Le explicará cómo funcionan los distintos mecanismos y le mostrará cómo implementarlos mediante una serie de ejercicios prácticos. Al final del curso, será capaz de escribir programas en C robustos y portables.

OBJETIVOS PEDAGÓGICOS

Al término de la formación, el alumno podrá:

Dominar la cadena de producción de un programa escrito en lenguaje C

Utilizar los operadores, expresiones y estructuras de control del lenguaje C

Manipulación de estructuras de datos, matrices, punteros y cadenas de caracteres

Organización del código del programa mediante funciones

Utilizar las principales bibliotecas estándar de C

Se pondrán a disposición de los participantes máquinas Unix o Windows (PC) para que puedan poner en práctica los conceptos presentados.

PROGRAMA

última actualización: 05/2024

1) Primeros pasos en C

- Introducción al lenguaje C y sus ventajas.
- C++ frente a C. Normas C++11 y C11.
- Archivos fuente (.c, .h).
- Estructura general de un programa.
- La sintaxis básica del lenguaje.
- Tipos de datos y constantes básicas.
- Variables globales y locales.
- Almacenamiento y paso de parámetros.
- Entradas/salidas formateadas.
- Los comentarios.
- Uso básico de la línea de producción.
- Entornos de edición, compilación y ejecución.
- Ejecutar un programa inicial.

2) Operadores y expresiones

- Operadores aritméticos.
- Mecanismos de evaluación de expresiones.
- Post y preincremento de decremento.
- Precedencia y asociatividad de los operadores.
- Operadores de asignación.
- Cómo funcionan las expresiones lógicas.
- Expresiones lógicas en sentencias while, if...
- Operadores de comparación : <, >, ==, !=...
- Operadores lógicos: AND, OR, negación.

PARTICIPANTES

Desarrolladores, ingenieros, jefes de proyecto cercanos al desarrollo.

REQUISITOS PREVIOS

Conocimientos básicos de programación.

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.

- Tipos numéricos compuestos. Regla de conversión en expresiones mixtas. Conversiones implícitas/explicitas.
- Inicialización de variables.
- Aritmética de direcciones.
- Formatos de entrada/salida asociados a los tipos digitales.
- Operadores bit a bit: AND, OR, OR exclusivo, complemento a 1, negación. Operadores de desplazamiento: <>, <<, >>.
- Expresión condicional con el operador ternario.

Trabajo práctico : Utilización de operadores y expresiones.

3) Estructuras de control

- Noción de bloques.
- Estructuras de bucle: while, for.
- Instrucciones de control de bucle: break, continue.
- Estructuras de elección: if, else, else if.
- Estructura de opción múltiple: interruptor.

Trabajo práctico : Implantación de estructuras de control.

4) Matrices, punteros y cadenas de caracteres

- Definición, inicialización y acceso a los elementos de un array.
- Definición de un puntero. Recuperación de la dirección de memoria de un objeto. Acceso al contenido de un puntero.
- Equivalentes de puntero/tabla.
- Cálculo de punteros.
- Cadenas de caracteres.
- Ejemplos de manipulación de cadenas.
- C11 Cadenas de caracteres Unicode.

Trabajo práctico : Manipulación de matrices, punteros y cadenas de caracteres.

5) Las estructuras

- Intereses de las estructuras.
- Declarar, inicializar y acceder a los campos de una estructura.
- Utilizar estructuras anidadas.
- Crear nuevos tipos utilizando Typedef.
- Campos de bits.
- Sindicatos.
- Enumeraciones.
- Estructuras anónimas y enumeraciones en C11.
- Definir punteros a estructuras.

Trabajo práctico : Implementación de nuevas estructuras de datos.

6) Las funciones

- Definición de una función.
- Llama a una función.
- Paso de parámetros: por valor o por referencia.
- Código de retorno de la función. Tipos de retorno.
- La función "principal".

Trabajo práctico : Descomposición del código mediante funciones. Gestión de las llamadas a funciones.

7) Compilación separada, clase de asignación

- Mecanismo de funcionamiento de la línea de producción.
- Utilización de bibliotecas de fuentes.
- Noción de Makefile.
- Configuración de la memoria de un programa C (pila, montón, etc.).
- Clases de asignación de variables (auto, register, static, extern).
- Diferentes escenarios para una compilación separada.

- Noción de objeto externo.
- Datos globales y estáticos.
- Datos locales.
- Regla de visibilidad.
- Información adicional sobre funciones e inicializaciones.

8) El preprocesador

- Utilizar macros predefinidas (constantes simbólicas). Definir sus propias macros con `#define`.
- Definición de macros como funciones. Uso de marcadores `#` y `##`.
- Anula la definición de la constante con `#undef`.
- Compilación condicional: `#if`, `#ifdef`, `#ifndef`, `#elif`, `#endif`.
- Incluya recursos con `#include`.

Trabajo práctico : Utilización de directivas de preprocesador. Configuración de la compilación condicional.

9) Bibliotecas estándar

- Funciones de cálculo matemático (`sqrt`, `sin`...).
- Funciones de entrada/salida (`fprintf`, `fscanf`...).
- Funciones de acceso a ficheros (`fread`, `fwrite`...).
- Funciones de manipulación de cadenas (`strlen`, `strcat`...).
- Funciones de gestión de memoria (`malloc`, `free`...).
- Configuración de estructuras encadenadas (listas encadenadas, árboles n-arios, etc.).
- Las funciones "seguras" de la biblioteca estándar C11 (`strcat_s`, `strlen_s`, etc.).

Trabajo práctico : Utilización de las principales funciones de la biblioteca estándar.

FECHAS

Contacto