

Introducción a la programación

PRESENTACIÓN

El dominio de un lenguaje de programación implica el manejo de ciertas técnicas comunes, como la programación estructurada, orientada a objetos y/o eventos, así como el uso de variables, distintos tipos de datos, eventos, clases, objetos, métodos y propiedades, como un requisito para adentrarse con cierta solidez en las características especiales de cada lenguaje de programación.

PERFIL DE INGRESO

El curso está dirigido a personas interesadas en resolver algoritmos directamente traducibles a un lenguaje de programación. Se requiere haber acreditado o demostrar conocimientos equivalentes al curso de Introducción a la computación e Internet con WINDOWS.

OBJETIVO

El participante resolverá algoritmos directamente traducibles a lenguajes de programación, mediante las técnicas de programación estructurada u orientada a objetos.

TEMARIO

1. LENGUAJES DE PROGRAMACIÓN
 - 1.1 Definición de lenguaje de programación.
 - 1.2 Definición de programa.
 - 1.3 Paradigmas de la programación.
 - 1.3.1 Programación funcional y estructurada.
 - 1.3.2 Programación orientada a objetos.
 - 1.3.3 Programación lógica.
 - 1.4 Traductores.
 - 1.4.1 Intérpretes.
 - 1.4.2 Compiladores.
 - 1.5 Código fuente, objeto y ejecutable.
 - 1.5.1 Depurador.
 - 1.6 Entornos de desarrollo integrado.
2. ELEMENTOS BÁSICOS DE LA PROGRAMACIÓN ESTRUCTURADA
 - 2.1 Tipos de datos primitivos.
 - 2.1.1 Numéricos (enteros y de punto flotante o reales).
 - 2.1.2 Símbolos y cadenas.
 - 2.1.3 Lógicos o booleanos (falso, verdadero).
 - 2.2 Palabras reservadas.
 - 2.3 Variables.
 - 2.4 Constantes.

- 2.5 Expresiones.
 - 2.5.1 Operadores y operandos.
 - 2.5.1.1 Operadores unarios.
 - 2.5.1.2 Operadores binarios.
 - 2.5.1.3 Operadores especiales.
 - 2.6 Jerarquía de operadores.
 - 2.6.1 Precedencia de operadores.
 - 2.6.2 Reglas de evaluación de expresiones.
 - 2.7 Operadores aritméticos.
 - 2.7.1 Suma.
 - 2.7.2 Resta.
 - 2.7.3 Multiplicación.
 - 2.7.4 División.
 - 2.7.5 Módulo.
 - 2.7.6 Potencia.
 - 2.8 Operadores relacionales.
 - 2.8.1 Mayor que.
 - 2.8.2 Menor que.
 - 2.8.3 Mayor o igual que.
 - 2.8.4 Menor o igual que.
 - 2.8.5 Igual a.
 - 2.8.6 Diferente de.
 - 2.9 Operadores lógicos o booleanos (and, or, not, xor).
 - 2.9.1 Tablas de verdad.
- 3. CONTROL DE FLUJO DE SENTENCIAS
 - 3.1 Sentencias incondicionales.
 - 3.1.1 Asignación.
 - 3.1.2 Lectura.
 - 3.1.3 Escritura.
 - 3.1.4 Transferencia incondicional de secuencia (go to).
 - 3.2 Sentencias condicionales.
 - 3.2.1 Selección (if).
 - 3.2.2 Ciclos (while, for).
 - 3.2.3 Selección múltiple (switch o select - case).
- 4. VARIABLES CON SUBÍNDICE O ARREGLOS
 - 4.1 Definición.
 - 4.2 Arreglos unidimensionales.
 - 4.3 Arreglos bidimensionales.
- 5. FUNCIONES
 - 5.1 Concepto de función en programación.
 - 5.2 Llamada o invocación a una función.
 - 5.3 Parámetros.
 - 5.3.1 Parámetros por valor.
 - 5.3.2 Parámetros por referencia.
 - 5.4 Valor de retorno.
- 6. ESTRUCTURAS DE DATOS

- 6.1 Listas.
- 6.2 Pilas.
- 6.3 Árboles.
- 7. ELEMENTOS BÁSICOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS
 - 7.1 Concepto de objeto.
 - 7.2 Anatomía de un objeto.
 - 7.3 Beneficios de la programación orientada a objetos.
- 8. CLASES Y OBJETOS
 - 8.1 Definición de una clase.
 - 8.2 Miembros de una clase.
 - 8.2.1 Propiedades.
 - 8.2.1.1 Métodos.
 - 8.2.1.2 Constructores y creación de objetos.
 - 8.2.1.3 Acceso a propiedades y métodos.
 - 8.2.1.4 destructores.
- 9. ENCAPSULAMIENTO
 - 9.1 Modularidad.
 - 9.2 Ocultamiento de la implementación.
 - 9.3 Protección de variables y métodos.
 - 9.3.1 Miembros privados.
 - 9.3.2 Miembros públicos.
 - 9.3.3 Miembros protegidos.
- 10. HERENCIA
 - 10.1 Jerarquía de clases.
 - 10.1.1 Clase raíz.
 - 10.1.2 Clase hoja.
 - 10.1.3 Subclases y superclases.
 - 10.2 Tipos de herencia.
 - 10.2.1 Herencia para reutilizar la implementación.
 - 10.2.2 Herencia por diferencia.
 - 10.2.3 Herencia para sustitución de tipos.

PERFIL DE EGRESO

Los conocimientos adquiridos permitirán al participante crear algoritmos con la finalidad de dar solución a problemas de distintos campos de aplicación, que sean fácilmente traducibles a algún lenguaje de programación.

REQUISITOS ACADÉMICOS

Estudios mínimos de preparatoria o equivalente.

DURACIÓN

30 horas.

Enero 2013