

CURSO-TALLER DE PROGRAMACIÓN "DISEÑO CON OBJETOS USANDO C++"

OBJETIVO

- * Obtener un buen entendimiento de conceptos de diseño y programación con objetos.
- * Entender el diseño de programas con objetos a través de código en C++.
- * Aprender los postulados de C++ y la manera de usarlos.
- * Familiarizarse con programación práctica con C++.
- * Identificar las fallas y malas prácticas en el desarrollo de programas con objetos
- * Recibir guías de estilo para codificar en C++
- * Enfocar los conceptos anteriores a la solución de problemas aplicados en las ramas de los negocios y las ingenierías.

A QUIÉNES VA DIRIGIDO

A los diseñadores y programadores de programas de aplicaciones o de sistemas que ya estén familiarizados con la programación en C. No se requiere experiencia previa con programación orientada al objeto.

FORMA DE IMPARTIRSE

El curso consistirá de cuatro días completos de exposición y ejercicios. Se alternarán las sesiones de estudio del diseño con objetos y del lenguaje C++, con otras donde los participantes trabajarán con programas en C++ en varios ejercicios cada vez más complejos. A través del uso de las técnicas de diseño y programación con objetos, los participantes podrán evaluar los beneficios de la tecnología de objetos con respecto a sus necesidades y aplicaciones. Se entregarán notas, un diskette conteniendo todos los programas en C++ y ejercicios, y un libro reciente de McGrawHill o Addison-Wesley sobre C++, al inicio del evento.

DURACIÓN

Cuatro días, ocho horas diarias, incluyendo sesiones de programación y ejercicios.

EJERCICIOS PRÁCTICOS

Habrà varios ejercicios donde el participante escribirà programas usando una estación de trabajo o una computadora personal (IBM PC o compatible). Estos ejercicios cubrirán los siguientes conceptos: * Lectura crítica de código y su corrección * Escribir código nuevo desde el principio * Modificar programas para satisfacer nuevos requisitos * Reuso de código * Integración de código.

Los participantes que así lo deseen pueden traer en un diskette (de 5¼ o de 3½") una copia de su editor de textos preferido.

Contenido

1. Diseño con objetos

1.1. Revisión de los principales métodos de diseño de programas

- 1.1.1. De arriba hacia abajo
- 1.1.2. De abajo hacia arriba
- 1.1.3. Comenzando por los objetos centrales

1.2. Metas del diseño con objetos

- 1.2.1. Reuso y maleabilidad al cambio

1.3. Alta productividad y confiabilidad

1.4. Abstracción de datos

- 1.4.1. Ocultamiento de información
Interfaces
- 1.4.2. Tipos de datos abstractos

1.5. Herencia

- 1.5.1. Jerarquías de clases
- 1.5.2. Polimorfismo

1.6. Diseñando para poder cambiar fácilmente

- 1.6.1. Diseño con muchas componentes rígidas y unas pocas flexibles
Determinando cuales componentes deben dejarse flexibles. ¿Cuánta flexibilidad es necesaria?
- 1.6.2. Identificación de riesgos
- 1.6.3. Prototipo
- 1.6.4. Guías para diseño práctico con objetos

2. Comparación entre C y C++

2.1. Conveniencias sintácticas

Comentarios. Declaraciones dentro de los bloques.

2.2. Constantes

Escalar. Apuntador. a una constante.

2.3. Funciones

- 2.3.1. Argumentos por omisión

3. Clases en C++

- 2.3.2. Argumentos variables
- 2.3.3. Sobrecarga de funciones
- 2.3.4. Prototipos y verificación de tipos
- 2.3.5. Funciones en línea

2.4. Tipos que son referencias

Variables. Parámetros. Inicialización. Uso.

3. Clases en C++

3.1. Introducción

Miembros que son datos (propiedades). Miembros que son funciones. Ranuras. Propiedades (miembros) privadas. Propiedades públicas. Amigos.

3.2. Llamadas implícitas

- 3.2.1. Constructores
- 3.2.2. Destruyores

3.3. Miembros

- 3.3.1. Miembros estáticos
- 3.3.2. Miembros constantes (const)

4. Herencia en C++

4.1. Introducción

- 4.1.1. Clases derivadas
- 4.1.2. Funciones virtuales

4.2. Llamadas implícitas

- 4.2.1. Constructores de base
- 4.2.2. Destruyores virtuales
- 4.2.3. Orden de llamadas

4.3. Clases base abstracta

Funciones virtuales puras

4.4. Visibilidad

Derivación pública. Miembros protegidos. Control de acceso

5. Manejo de memoria en C++

5.1. Asignación de memoria

6. Temas avanzados de C++

5.1.1. Introducción

Funciones new y delete.

5.1.2. Objetos estáticos y dinámicos

Escalares. Arreglos. Inicialización. Automatic. Pila.

5.2. Estrategias para asignar memoria

5.2.1. Redefinición de las funciones new y delete

a) En todo el programa. b) Por cada clase.

6. Temas avanzados de C++

6.1. Funciones en línea

Funciones miembro. Constructores

6.2. Flujo de control

Cuándo se invocan los destructores

6.3. Algunas áreas confusas

6.3.1. Inicialización y asignación

6.3.2. Espacios de nombres de los identificadores

6.3.3. Alcance o rango (scope) de una variable

6.3.4. Regresando objetos como valor o resultado

6.3.5. Los objetos como miembros

6.4. Características actuales del lenguaje

Herencia múltiple. Clases base virtuales. Sobrecarga de operadores.

6.5. Características propuestas

Manejo de excepciones. Tipos parametrizados.

7. Organización y compilación de programas en C++

7.1. Organización de los programas

Archivos de encabezados (headers). Convenciones para nombrar los archivos. Bibliotecas

7.2. Construcción de ejecutables

Dependencias. Eslabonamiento de tipos a prueba de errores. Eslabonamiento con código C.

7.3. Mensajes de error

8. Otros lenguajes para objetos

Smalltalk. C-objetivo. Eiffel.