

CURSO-TALLER "DISEÑO CON OBJETOS USANDO C++ PARA QUIENES DESCONOCEN C"

OBJETIVO

- * Aprender los postulados de C++, sin saber C.
- * Entender el diseño con objetos a través de programas concretos en C++
- * Tener una perspectiva amplia de los conceptos de la programación con objetos
- * Familiarizarse con programación práctica con C++.
- * Desarrollar guías para el diseño con objetos
- * 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 conozcan otros lenguajes de programación basados en procedimientos, tales como Cobol, Pascal, o Fortran. No se requiere conocer C, ni experiencia previa con programación usando objetos. También pueden beneficiarse gerentes y administradores interesados en los detalles técnicos de la programación con objetos.

FORMA DE IMPARTIRSE

Durante cuatro días, los participantes entenderán la naturaleza dual de C++: alto nivel de abstracción de datos y programación con objetos, junto con los detalles de un lenguaje de bajo nivel. Habrá sesiones de diseño con objetos; del lenguaje C++; de trabajo en la computadora con varios programas en C++. El subconjunto de C++ a presentarse es suficiente para aprender las bases de programación con objetos, incluyendo clases, herencia, y polimorfismo. Los principios del diseño con objetos se presentarán y relacionarán con las correspondientes características del lenguaje. Se entregarán notas, un diskette conteniendo todos los programas en C++ y ejercicios, y un libro reciente de McGraw Hill 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 escribirá 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. Conceptos sobre objetos en C++

1.1. Metas del diseño con objetos

- 1.1.1. Reuso y maleabilidad al cambio
- 1.1.2. Alta productividad y confiabilidad

1.2. Abstracción de datos

- 1.2.1. ¿Qué es un objeto?
- 1.2.2. Interfaces
- 1.2.3. Encapsulación
- 1.2.4. Instanciación
- 1.2.5. Tipos de datos abstractos

1.3. Abstracción de datos en C++

- 1.3.1. Clases
- 1.3.2. Datos miembros (propiedades y ranuras)
- 1.3.3.

1.4. Herencia

- 1.4.1. Derivación de clases.
- 1.4.2. Jerarquía "es un".
- 1.4.3. Clases derivadas.
- 1.4.4. Jerarquías de herencia

1.5. Polimorfismo

- 1.5.1. Sobrecarga de funciones.
- 1.5.2. Liga dinámica.
- 1.5.3. Funciones miembro virtuales.

2. Diseño con objetos

2.1. El proceso de diseño

- 2.1.1. Diseño incremental.
- 2.1.2. Identificación de objetos

2.2. Notación para diseño

- 2.2.1. Diagramas de clase

2. Diseño con objetos

2.2.2. Diagramas de objetos.

2.3. Criterios para un diseño bien fundado

2.3.1. Acoplamiento.

2.3.2. Cohesión.

2.4. Métodos de diseño

2.4.1. Generación de interfaces

2.4.2. Uso de bases de datos

2.4.3. Encapsulación por medio de API's más apropiadas

2.4.4. Uso de estándares o de interfaces populares

3. Introducción al C++

3.1. Programas sencillos

3.1.1. Tipos de datos inter-construidos.

3.1.2. Expresiones.

3.1.3. Flujo de control.

3.1.4. Entrada y salida básica por un caudal de datos (stream).

3.2. Funciones

3.2.1. Argumentos

3.2.2. Resultados.

3.2.3. Prototipos.

3.3. Estructuras de datos

3.3.1. Arreglos.

3.3.2. Cadenas.

3.3.3. Apuntadores.

3.3.4. Estructuras de datos comunes.

3.4. Asignación dinámica de memoria

3.4.1. Memoria estática versus dinámica.

3.4.2. Memoria automática versus de la pila.

3.4.3. Los operadores new y delete.

4. Clases

4.1. Introducción

4.1.1. Miembros dato

4. Clases

4.1.2. Miembros funciones (métodos)

4.2. Encapsulación

4.2.1. Miembros públicos.

4.2.2. Miembros privados

4.2.3. Miembros protegidos.

4.3. Herencia

4.3.1. Herencia pública

4.3.2. Funciones virtuales

4.3.3. Clases base abstractas

4.3.4. Funciones virtuales puras

4.4. Inicialización

4.4.1. Constructores

4.4.2. Destrucción

4.4.3. Destrucción virtuales

4.5. Estado compartido

4.5.1. Miembros dato estáticos

4.5.2. Miembros función estáticos

5. Conceptos avanzados de C++

5.1. Características esenciales

5.1.1. Asignación versus inicialización

5.1.2. Arreglos de objetos

5.1.3. Sobrecarga de funciones

5.1.4. Sobrecarga de operadores

5.1.5. Tipos referencia

5.1.6. Tipos constante

5.1.7. Argumentos por omisión

5.1.8. Asignación dinámica de memoria

5.2. Características avanzadas

5.2.1. Herencia múltiple.

5.2.2. Tipos parametrizados

5.2.3. Excepciones

5.3. Otros lenguajes para objetos

Objective C. Smalltalk. Eiffel