

# INTRODUCCIÓN A C++

Catedrático: Ing. David Rajo

[davidraxo@gmail.com](mailto:davidraxo@gmail.com)

# INTRODUCCIÓN A C++

- C++ es un lenguaje de programación de alto nivel y de propósito general, y es una extensión de C con características mas potentes.
- Un lenguaje de programación es un idioma artificial diseñado para expresar procesos que pueden ser llevadas a cabo por máquinas.
- Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones

# Código Fuente

- El proceso de traducir un algoritmo en pseudocódigo o diagrama de flujo a un lenguaje de programación se denomina codificación, y el algoritmo escrito en un lenguaje de programación se denomina **código fuente**.
- La computadora en realidad no comprende los lenguajes de programación, debe traducirlos a **código maquina** que es un lenguaje que si entiende la maquina directamente

# Estructura general de un programa

- Directivas
- Archivos de cabecera
- Declaraciones Globales
- La función `main()`;
- Funciones definidas por el usuario
- Comentarios de programa
- Sentencias

# Directiva #include y archivos de cabecera

- #include

indica al compilador que lea el archivo fuente que viene a continuación de ella y su contenido lo inserte en la posición donde se encuentra dicha directiva.

- Archivos de cabecera

(archivos con extensión .h o .hpp contienen código fuente C/C++) se sitúan en un programa C++ mediante la directiva del preprocesador #include con una instrucción.

Los archivos tales como `iostream` se denominan archivos de inclusión, archivo, normalmente en forma de código fuente,.

# Directiva using y using namespace std;

- Si utiliza el estándar ANSI/ISO C++ en lugar de `iostream.h`, propio de las versiones antiguas de C++, deberá utilizar la directiva `using` de espacio de nombres para permitir que las definiciones de `iostream` estén disponibles en su programa.
- **Un espacio de nombre** es una parte del programa en el cual se recuerdan ciertos componentes que son desconocidos o no son reconocidos.
- **Using namespace std;** indica que todas las sentencias del programa que vienen a continuación están dentro del espacio de nombres `std`. Diversos componentes como `cout` están declarados dentro de este espacio de nombre.
- **Nota:** Si no se utiliza la directiva se necesitará preceder el nombre `std` a muchos elementos del programa.

# Función main()

- Cada programa de C++ debe contener una función main() la cual es la función principal y el cual es el primer punto de entrada de un programa. Cuando se ejecuta el programa, se invoca en primer lugar a la función main()

```
int main(){  
    Sentencias;  
    Return 0;  
}
```

El cuerpo de la función es un conjunto de sentencias que se ejecutan cuando se ejecuta main()

**Nota:** Las sentencias terminan con punto y coma “ ; ”.

**Return 0;** termina la función main

# Sentencia

- Una Sentencia representa una instrucción completa a la computadora. Cada sentencia se debe terminar con punto y coma.



# Declaraciones Globales

- Las declaraciones globales indican al compilador que las funciones definidas por el usuario o variables así declaradas son comunes a todas las funciones de su programa. Las declaraciones globales se sitúan antes de la función `main()`.

# Funciones definidas por el usuario y funciones de biblioteca

- Todos los programas en C++ se construyen a partir de funciones
- **Función** como idea general, se presenta como un subalgoritmo que forma parte del algoritmo principal, el cual permite resolver una tarea específica. Algunos lenguajes de programación, como Visual Basic .NET o Fortran, utilizan el nombre función para referirse a subrutinas que devuelven un valor.
- Mientras que un tipo son definidas por el usuario las otras se encuentran definidas en bibliotecas del lenguaje de programación y pueden ser invocadas gracias a los archivos de cabecera

# Entrada/Salida por consola

- Entrada (cin)
- Salida (cout)

Operadores de extracción e inserción, >> y <<, apuntan en la dirección del flujo de datos

Ejemplo

# Ejemplo de Estructura

```
#include <iostream>
using namespace std;
char materia[20] = "Programacion I";
int main(){
    int p1,p2,pf,nf;
    p1 = 8.1;
    p2 = 7;
    pf = 0;
    nf = p1*0.3 + p2*0.3 + pf*0.4; // calcula la nota final
    cout << "Materia: " << materia << endl;
    cout << "Nota Final: " << nf << endl;
    return 0;
}
```

# Traductores de Lenguaje

- Son programas que traducen código fuente escrito en un lenguaje de programación de alto nivel a código maquina.
- Los traductores se dividen en compiladores e interpretes

# Interpretes y compiladores

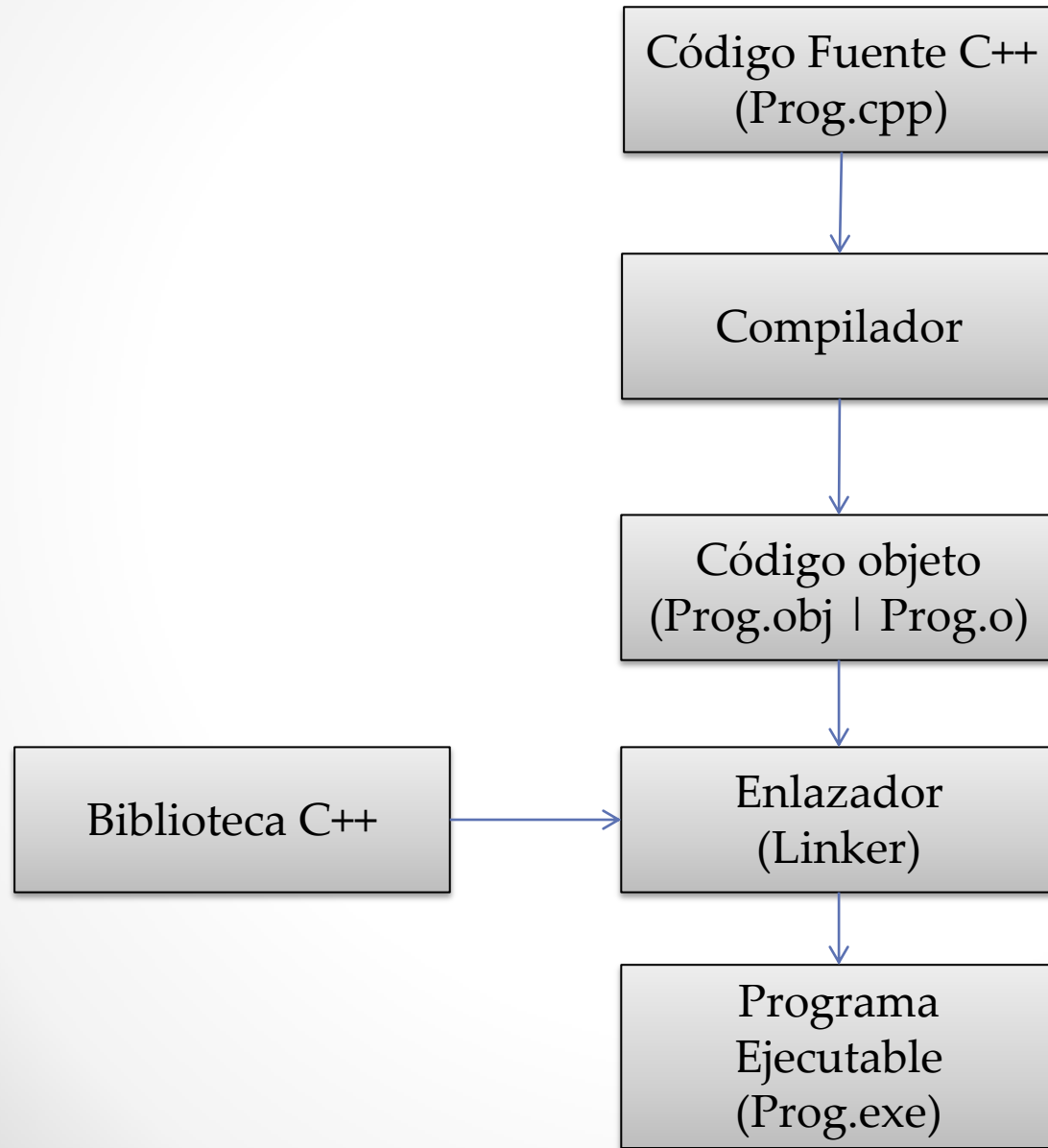
- Interprete

Es un traductor que toma un programa fuente, lo traduce y, a continuación lo ejecuta. Este proceso se hace línea a línea.

- Compilador

Es un programa que traduce los programas fuente a lenguaje maquina. La traducción del programa completo se realiza en una sola operación denominada compilación del programa

# Construcción de un programa en C++



# Tipos de Compiladores

- Compiladores en Línea de comandos
- Compilador incluido en un Entorno Integrado de Desarrollo (IDE)

Las etapas de compilación y enlace se realizan automáticamente por el compilador

Las IDE son sistemas de programación que ayudan al desarrollador con diferentes herramientas como Editor, Compilador, Enlazador, Gestor de proyectos, Depurador y otras herramientas integradas en un paquete de software



# Pasos para Compilar un programa de C++

- Los pasos para la compilación del programa son básicamente 4:
- Abrir **Símbolo del sistema**
- Agregar la Ruta (PATH) a la variable de entorno  
**"PATH=%PATH%;C:\Program Files (x86)\CodeBlocks\MinGW\bin"**  
(este paso se puede evitar agregando la ruta a la variable de entorno de forma permanente)
- Buscar la carpeta en la que se encuentra el programa escrito y guardado con la extensión **cpp** (**cd c:\ejemplos\ejemplo1**)
- Utilizar el comando que nos sirve para compilar el archivo (**c++ holamundo.cpp -o holamundo.exe**)

# Caso Práctico de Compilación

- Compilar en línea de comandos utilizando el compilador MinGW.

Utilice la línea de ordenes:

```
c++ -g -Wall -o hola hola.cpp
```

Mas simple

```
c++ hola.cpp
```

## Opciones

- g: para permitir la depuración
- Wall: muestra todos los mensajes de error y advertencia del compilador
- o hola: indica el nombre del archivo de salida, cualesquiera sean las etapas cumplidas
- c: realiza preprocesamiento y compilación, obteniendo el archivo en código objeto; no realiza el enlazado.

# Resumen

Para producir un ejecutable con fuente de un solo archivo:

✓ `c++ hola.cpp -o hola.exe`

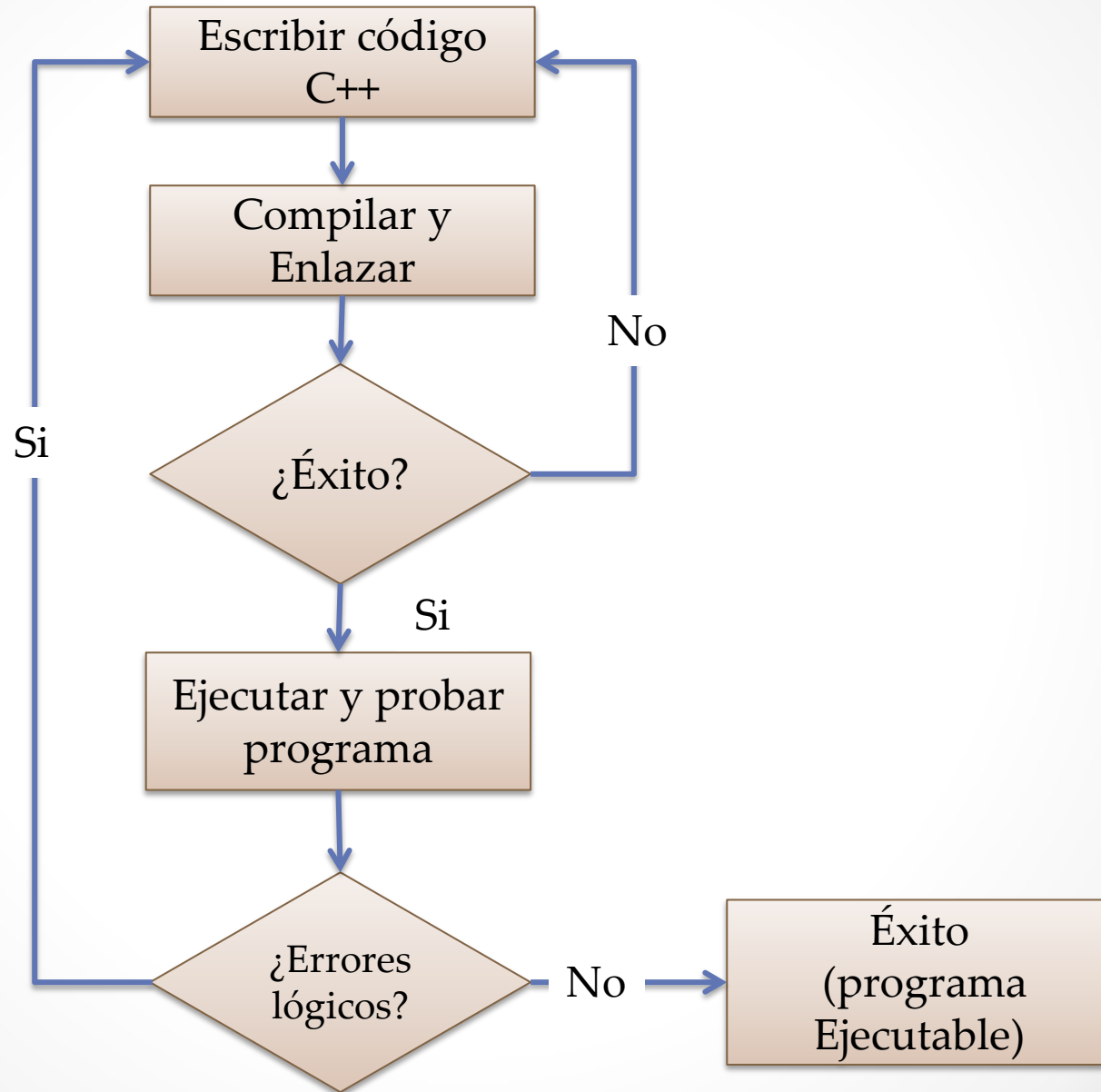
Para crear un módulo objeto, con el mismo nombre del fuente y extensión .o:

✓ `c++ hola.cpp -c`

Para enlazar un módulo objeto:

✓ `c++ -o hola hola.o`

# Depuración de un Programa en C++



# Tipos de Errores

- Errores de Sintaxis
  - Son los que se producen cuando el programa viola la sintaxis, es decir las reglas de gramática del lenguaje.
- Errores Lógicos
  - Representa errores del programador en el diseño del algoritmo y posterior programa.
- Errores de Regresión
  - Son los que se crean accidentalmente cuando se intenta corregir un error lógico.
- **Depurador** (debugger) programa diseñado específicamente para la detección, verificación y corrección de errores

# El operador sizeof

- Se utiliza para conocer el tamaño en bytes de un tipo de dato o variable.
- Ejemplo

# Conversión de Tipos

- Conversión Implícita
- Conversión Explícita
  
- Ejemplo

# Aplicaciones

1. Crear programa que determine la edad de una persona por el año de nacimiento
2. Teniendo como datos de entrada el radio y la altura de un cilindro queremos calcular: el área lateral y el volumen del cilindro.
3. Crear un programa que resuelva la ecuación cuadrática
4. Escribir un programa simule un ATM para que solicite al usuario una cantidad en dólares y transforme la cantidad en número de billetes de cada denominación que necesita para formarla