Introducción al C++

Mag. Ing. Nancy López

El lenguaje C/C++

- El lenguaje de programación C++ se comenzó a desarrollar en 1980 en los laboratorios de la Cía. AT&T. Es una ampliación del lenguaje C.

- El nombre C++ proviene del operador incremento ++.

Aspectos básicos

- Los archivos fuente tienen la extensión **.cpp** (de C plus plus)
- Los encabezados tienen la extensión .h y se incluyen mediante la directiva **#include** (Ej. #include <iostream>)
- Permite declarar las variables en cualquier parte del programa, pero la programación estructurada indica que se deben declarar al inicio del programa.

Aspectos básicos

- Los comentarios se pueden hacer de dos formas:
- Una sola línea: // este es un comentario de una línea
- Un bloque: /* comentario de un bloque.
  Sigue en la siguiente línea */
- Las llaves **{..}** constituyen el modo utilizado en el lenguaje C++ para agrupar las funciones o los bloques.

Aspectos básicos

- Cada sentencia debe terminar en un “Punto y coma” (**;**), ejemplo: a=3*4;
- En una misma línea se pueden poner varias sentencias, separadas por el “;” pero se recomienda escribir una sentencia por línea.

  Por ejemplo: a=var1*5; b=var2*6;
- Existe una única función principal que está por encima de todas, por la cual se empieza la ejecución del programa. Dicha función es **main()**.

Aspectos básicos

- En algunos IDE se requiere utilizar int main() y al final del programa colocar un return N°;
- Ej. int main(){
  //resto del programa
  return 0; // return 1;
  }
- Se programa en minúsculas.
- Es case sensitive (diferencia mayúsculas de minúsculas)

  Por ejemplo, ana != Ana

La función main()

- La función **main()** es la función principal del programa, por donde empieza la ejecución del mismo.
- Como esta función no recibe argumentos y tampoco retorna ningún valor, se utiliza la palabra **void** (vacío) para indicarlo y los paréntesis vacíos: void main()
- Pero en Zinjal, como ya mencionamos tenemos que devolver un entero. **int main()**

Estructura general de un programa en C/C++

Encabezados

Función principal

{

  Declaración de variables

  Cuerpo del programa

}

Estructura general de un programa en C/C++

```
#include<iostream>  //archivo de encabezado
using namespace std;  //específico de Zinjai para poder usar la    biblioteca
    iostream
int main()    //función principal
{    //llave que indica inicio del programa
   cout<<“Primer programa”;   //salida por pantalla
  return 0;
}    //llave que indica fin del programa.
```

Tipos básicos de datos

- Entero: **int**
- Entero largo: **long int**
- Decimal: **float**
- Decimal: **double**
- Carácter: **char**
- Forma de declaración: **tipo nombre**

Ej.: int a, suma=0;
     char seguir;

Operadores

Operadores Aritméticos		
Operador	Utilización	Descripción
+	op1 + op2	Suma op1 y op2
-	op1 - op2	Resta op1 y op2
*	op1 * op2	Multiplica op1 y op2
/	op1 / op2	Divide op1 y op2
%	op1 % op2	Calcula el resto de dividir op1 por op2

Operadores Relacionales		
Operador	Utilización	Devuelve true si:
>	op1 > op2	op1 es mayor que op2
>=	op1 >= op2	op1 mayor o igual que op2
<	op1 < op2	op1 es menor que op2
<=	op1 <= op2	op1 es menor o igual que op2
==	op1 == op2	op1 es igual a op2
!=	op1 != op2	op1 distinto de op2

Operadores

Otros operadores		
Operador	**Utilización**	**Descripción**
&& , ||	op1&&op2	Oper. Lógicos AND y OR
++ , --	var1++ , var1–	Operador incremental y decremental
? :	expres ? op1:op2	Operador de comparación terciario

Operadores de Asignación		
Operador	**Utilización**	**Descripción**
=	res = expresión	Operador de asignación simple
+=	op1 = op1 + val	Sumar y asignar
-=	op1 = op1 - val	Restar y asignar
***=**	op1 = op1 * val	Multiplicar y asignar
/=	op1 = op1 / val	Dividir y asignar

Conversión de tipos

- Cuando los operandos que intervienen dentro de una *expresión son de tipos diferentes (int,* long, float, etc), éstos se convierten automáticamente a un solo tipo común, que por lo general es el de mayor precisión. A este tipo de conversión se le denomina ***Conversión Implícita de tipos****.*

Por ejemplo:

```
int x=4; long z=3;
float y=2.5, ret;
ret = x * y * z; //ret=30.0
```

– Algunas de las reglas que rigen la conversión, aplicadas en el siguiente orden, son:

- Si un operando es **double, el otro es convertido a double, aunque sea entero (int, short o long).**
- Si un operando es **float, el otro es convertido a float.**
- Si un operando es **long, el otro es convertido a long.**
- En cualquier otro caso , ambos operados son del tipo **int.**

Conversión de tipos

Conversión Explícita: En C++ se dispone de una conversión explícita de tipos de variables, directamente controlada por el programador, llamada ***cast.***

- El ***cast se realiza anteponiendo al nombre de la variable o expresión el tipo de variable a la que se*** desea convertir, encerrado entre paréntesis. Por ejemplo:

```
float y=4.5,z=2.1;
int x=(int)(y/z) + 1; // y/z=2.14 x=2+1=3
```

- Por último, se debe tener mucho cuidado con las conversiones implícitas, ya que muchas veces se pierde información sin poder notarlo, sobre todo cuando las expresiones son un poco extensas, en cuyo caso es más conveniente dividirla en sub-expresiones.

Declaración de constantes

- Se declara (define) con la directiva #define.
- Por convención, las constantes van con mayúsculas.

  Ej. #define PI 3.1416;

  #define CANT 500;

Entrada y salida de datos

- Para la entrada y salida de datos se usan flujos (o stream)
- Se usa la biblioteca iostream
- Los flujos son:
  - **cout** (consola out): flujo de salida a la pantalla
  - **cin** (consola in): flujo de entrada
- Se usan en conjunto con los operadores << y >>

Entrada y salida de datos

- Ejemplo:

```
#include<iostream>
using namespace std;
int main()
{
   int a;
   cout<<"Ingrese un valor ";
   cin>>a;
   cout<<"El doble del valor ingresado es: "<<a*2;
   return 0;
}
```

Entrada y salida de datos

- En el ejemplo anterior se muestra texto y valores de variables en la misma línea:

  cout<<"El doble del valor ingresado es: "<<a*2;

- Los símbolos “<<“ se usan para separar texto de valores.

Formato de entrada y salida de datos

- C/C++ cuenta con algunos manipuladores de flujos (usa <iomanip.h>):
  - **endl**: se imprime un '\n' y se vacía el buffer de salida.
  - **setw(int num)**: establece la anchura mínima de campo. Necesita declarar <iomanip.h>
  - **setprecision(p)**: establece el número de cifras