

Principios de C++

Elaborado por: Ukranio Coronilla

Introduzca en un archivo y con ayuda de un editor como vi, gedit o alguno de su preferencia, el programa 1-1 y guárdelo como programa1_1.cpp (la extensión del archivo para los programas en C++ debe ser cpp).

Programa 1-1

```
1 #include <iostream>
2 using namespace std;
3
4 int main( )
5 {
6     int numeroDeLenguajes;
7
8     cout << "!!!Hola!!!.\n" << "Bienvenido a C++.\n";
9     cout << "Cuantos lenguajes de programación dominas? ";
10    cin >> numeroDeLenguajes;
11
12    if (numeroDeLenguajes < 1)
13        cout << "Sería recomendable aprender antes un lenguaje más sencillo...\n" << "por ejemplo C, aunque nada es imposible.\n";
14    else
15        cout << "Este curso será sencillo para ti.\n";
16    return 0;
17 }
```

Posteriormente construya el programa ejecutable con la siguiente instrucción en la línea de comandos:

```
g++ programal_1.cpp -o programal_1
```

g++ es el compilador C++ más utilizado en los sistemas UNIX, la opción -o indica al compilador que el nombre del archivo ejecutable será el que se especifica a continuación (en este caso programa1_1). Se recomienda darle el mismo nombre al ejecutable que al código fuente para evitar confusiones, y es también habitual dejar el archivo sin una extensión.

Ejecute el programa anteponiendo “./” al nombre del ejecutable:

```
./programal_1
```

Descripción del funcionamiento

Las primeras dos líneas inicializan las librerías necesarias para el manejo de funciones, las cuales realizan operaciones de entrada y/o salida a la consola.

La línea 1 busca el archivo llamado `iostream` y lo incluye al inicio del programa, este archivo contiene básicamente la definición de las funciones.

En las líneas 8 y 9 dentro de comillas se encuentra texto conocido como **string** o cadena. La instrucción `cout` provoca la impresión de la cadena en la consola. También es posible que `cout` imprima el valor de cualquier variable. Se pueden concatenar cadenas y variables utilizando como conector el operador "`<<`". Un salto de línea puede escribirse también con la palabra reservada `endl`. Por ello las siguientes dos instrucciones son equivalentes:

```
cout << "\n";  
cout << endl;
```

Es recomendable usar un salto de línea al final del programa pues algunos sistemas pueden no imprimir la última línea, dejándola almacenada en el buffer de salida.

La línea 10 obtiene un valor introducido en la consola por el usuario y lo asigna a la variable `numeroDeLenguajes`. Observe que la dirección a la que apunta la flecha es opuesta a la de `cout`. También es posible recibir más de un valor, por ejemplo:

```
cin >> numeroDeLenguajes >> numeroDeDialectos;
```

Obtiene dos valores introducidos por el usuario desde la consola.

La línea 15 es forzosa en algunos compiladores y permite que el programa principal regrese un valor al intérprete de comandos, tal y como se había estipulado que lo haría la línea 4.

En muchas ocasiones se requiere declarar constantes en el programa, por ejemplo el valor de π , para ello se utilizaría la instrucción:

```
const double PI = 3.1415926;
```

donde es usual utilizar letras mayúsculas para denotar la constante dentro del programa y anteponer la palabra reservada `const` indicando que no cambiará.

Ejercicio 1: Elabore un programa que encuentre el área de un círculo declarando la constante π , el usuario debe proporcionar el radio. Intente modificar π dentro del programa, ¿Qué ocurre?

Ejercicio 2: Un error común difícil de detectar es dividir dos tipos de datos enteros o un entero con un tipo flotante. Elabore un programa para imprimir el resultado de una división entre dos enteros y de un entero con un flotante y determine cuál es el error que ocurre.

Ejercicio 3: Se ha elaborado el siguiente programa para convertir grados centígrados a Fahrenheit

```
double c = 20;  
double f;  
f = (9/5) * c + 32.0;
```

Pruébalo y en caso de estar mal, corríjalo para que realice la conversión de manera adecuada.

Una manera para resolver el ejercicio anterior es utilizar la conversión temporal de tipo o cast. Es decir podemos transformar de manera temporal un tipo de dato a otro con:

static_cast<Type>(Expression)

por ejemplo la variable entera `varInt` la podemos cambiar a `double` con:

```
static_cast<double>(varInt)
```

Ejercicio 4: Pruebe a resolver el ejercicio anterior mediante un cast.

Ejercicio 5: No es recomendable utilizar los operadores de incremento y decremento en expresiones largas, por ejemplo C++ no garantiza para la siguiente expresión `n + (++n)` y con `n = 2` si la expresión devuelve 5 o 6. Elabore un programa para probar la aseveración anterior.

Ejercicio 6: Imprima la variable `precio` después de hacer la siguiente asignación:

```
double precio 78.7;
```

suponiendo que se desea imprimir dos dígitos decimales puesto que se trata de dinero, tendrá que insertar las siguientes líneas:

```
cout.setf(ios::fixed);  
cout.setf(ios::showpoint);  
cout.precision(2);
```

verifique que funciona dentro del programa.

En el programa 1-1 se obtiene un valor numérico del usuario, sin embargo es posible obtener una cadena del usuario como lo muestra el programa 1 -2

Programa 1-2

```
1 #include <iostream>  
2 #include <string>  
3  
4 using namespace std;  
5  
6 int main( )  
7 {  
8     string nombre;    //Variable para almacenar el nombre  
9  
10    cout << "Cuál es tu nombre?" << endl;  
11    cin >> nombre;  
12    cout << "Hola " << nombre << endl;  
13  
14    return 0;  
15}
```

Observe que se debe añadir la librería `string` en la línea 2, y que también se tuvo que declarar una variable de tipo cadena llamada `nombre` en la línea 8. Note que `cin` solo lee el primer token.

Proyecto 1 para subir al MOODLE:

1.-Se desea calcular la raíz cuadrada de un número entero n utilizando al algoritmo babilónico. Investigue en Internet el algoritmo y prográmelo, haciendo n del tipo `double` y el resultado con cuatro decimales.

2.- Escriba un programa que imprima la altura de un edificio, si se conoce el tiempo que tardó una pelota de tenis desde que se dejó caer en la parte más alta del edificio y hasta llegar al piso.

3.- Escriba un programa que reciba del usuario un valor entero de tiempo en segundos, y que imprima su equivalente en horas, minutos y segundos. Por ejemplo si el usuario introduce 50381, entonces el programa debe imprimir: 13 horas, 59 minutos y 41 segundos.