

Expresiones y tipos

Programacion en C++

Taller de programación



Departamento de Informática
Universidad Técnica Federico Santa María

Un primer ejemplo

ejemplo.cpp:

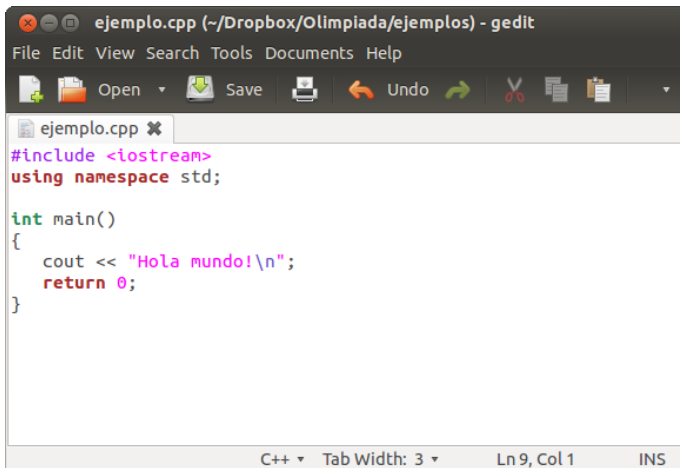
```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hola mundo!";
    return 0;
}
```

- ▶ `#include <iostream>` ← librería para entrada y salida
- ▶ `int main()` ← aquí **comienza** el programa
- ▶ `cout << ...` ← sentencia de salida
- ▶ `return 0;` ← aquí **termina** el programa

Editor de texto

Usaremos un editor de texto para escribir nuestro programa.
Gedit:



The screenshot shows the Gedit text editor window titled "ejemplo.cpp (~/.Dropbox/Olimpiada/ejemplos) - gedit". The menu bar includes File, Edit, View, Search, Tools, Documents, and Help. The toolbar contains icons for Open, Save, Print, Undo, and Redo. The editor displays the following C++ code:

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hola mundo!\n";
    return 0;
}
```

The status bar at the bottom indicates "C++", "Tab Width: 3", "Ln 9, Col 1", and "INS".

Compilación y ejecución

Para ejecutar el programa, primero tenemos que compilar el archivo. Para esto usaremos el **terminal**.

- ▶ Compilación:

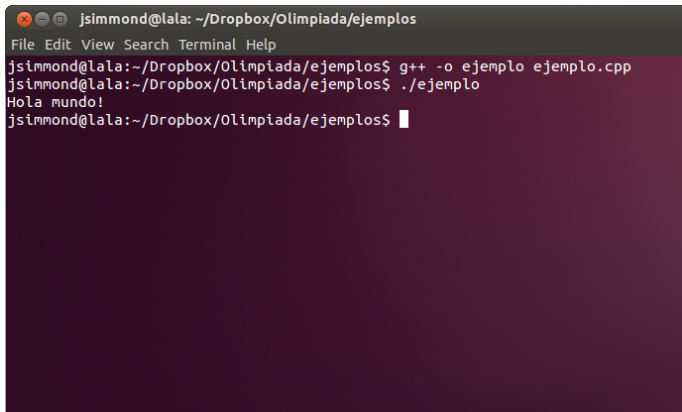
```
user@pc:~/Dropbox/Olimpiada/ejemplos\$ g++ ejemplo.cpp -o ejemplo
```

- ▶ Ejecución:

```
user@pc:~/Dropbox/Olimpiada/ejemplos\$ ./ejemplo
```

Editor de texto

Ejecutemos nuestro programa:



```
jsimmond@lala: ~/Dropbox/Olimpiada/ejemplos
File Edit View Search Terminal Help
jsimmond@lala:~/Dropbox/Olimpiada/ejemplos$ g++ -o ejemplo ejemplo.cpp
jsimmond@lala:~/Dropbox/Olimpiada/ejemplos$ ./ejemplo
Hola mundo!
jsimmond@lala:~/Dropbox/Olimpiada/ejemplos$
```

Instrucciones y bloques

- Cada instrucción debe terminar con un ;

```
x = y;  
y = y + 1;
```

Esto es equivalente a:

```
x = y; y = y + 1;
```

- Un bloque es una secuencia de instrucciones, delimitadas por { y }:

```
{  
    cout << "Hola mundo";  
    return 0;  
}
```

Comentarios

Pueden insertar comentarios en sus programas, para acordarse de que hacen las instrucciones:

- ▶ Pueden hacer comentarios de una línea

```
cout << "Hola mundo\n"; // imprime "Hola mundo" en el terminal
```

- ▶ Pueden comentar varias líneas:

```
/* ahora la instruccion de salida esta "escondida" del compilador  
cout << "Hola mundo\n"; // imprime "Hola mundo" en el terminal  
*/
```

Variables

- ▶ Usaremos variables para almacenar valores de diferentes tipos
- ▶ Sus características son:
 - nombre – un identificador válido
 - tipo – número entero (`int`), número real (`float`, `double`), carácter (`char`), etc.
 - valor – el valor almacenado
- ▶ En C++ es necesario definir todas las variables que se utilicen:

```
int a, b, c;  
float r, s;
```

- `a`, `b`, `c` son “declarados” como enteros, pero no tienen definido un valor inicial
- `r`, `s` son declarados como reales, tampoco tienen valor inicial

Asignación

- ▶ La operación básica para el uso de variables es la asignación
- ▶ El tipo de constante a asignar debe de ser congruente con el tipo con el que fue declarada la variable.
- ▶ Para la asignación se utiliza el símbolo =
- ▶ Ejemplo:

```
a = 5;  
b = 8;  
cout << a << b;
```

Imprime:

58

Ejemplo 2

```
#include <iostream>
using namespace std;

int main()
{
    int a, b, c, d; //4 variables enteras
    a = 45;    b = 23;
    c = 368;   d = 345;

    cout << "Impresion de \"variables\"\\n";
    cout << "a\\tb\\tc\\td\\n";
    cout << a << "\\t" << b << "\\t" << c << "\\t" << d << endl;
    return 0;
}
```

Ejemplo 2

```
#include <iostream>
using namespace std;

int main()
{
    int a, b, c, d; //4 variables enteras
    a = 45;    b = 23;
    c = 368;   d = 345;

    cout << "Impresion de \"variables\"\\n";
    cout << "a\\tb\\tc\\td\\n";
    cout << a << "\\t" << b << "\\t" << c << "\\t" << d << endl;
    return 0;
}
```

Resultado:

```
jsimmond@lala:~/Dropbox/Olimpiada/ejemplos$ ./ejemplo2
Impresión de "variables"
a      b      c      d
45     23     368    345
jsimmond@lala:~/Dropbox/Olimpiada/ejemplos$
```

Ejercicios

- ▶ Escriba un programa que genere la siguiente salida:

```
*  
**  
***  
****  
*****
```

- ▶ Escriba un programa que imprima la siguiente tabla. Utilice tabuladores.

Color	R	G	B
Rojo	255	0	0
Verde	0	255	0
Blanco	255	255	255
Gris	128	128	128

- ▶ Escriba un programa que declare las variables a y b como enteras y las variables c y d como reales. Asigne valores aceptables a cada variable. ¿Es posible asignar una variable entera a una real? ¿Es posible asignar una variable real a una entera? Escriba código para probar lo anterior y asigne diferentes valores.

Operadores aritméticos

Si `int a = 10, b = 20;`

Operación	Operador	Ejemplo	Resultado
Suma	+	<code>a + b</code>	30
Resta	-	<code>a - b</code>	-10
Multiplicación	*	<code>a * b</code>	200
División	/	<code>b / a</code>	2
Modulo	%	<code>b % a</code>	0
Incremento	++	<code>a++</code>	11
Decremento	--	<code>a--</code>	9

Reglas de precedencia

El operador `()` se utiliza para modificar la precedencia de los operadores.

Operador	Operación	Precedencia
<code>()</code>	Paréntesis	precedencia mas alta, se evalúan primero los más anidados
<code>*</code> , <code>/</code> , <code>%</code>	Multiplicación, división, modulo	se evalúan de izquierda a derecha
<code>+</code> , <code>-</code>	Suma, resta	se evalúan de izquierda a derecha

Ejemplos

Notación estándar	Notación en C++
$a + b - \frac{d}{c}$	<code>a + b - d/c</code>
$a \bmod c + d$	<code>a % c + d</code>
$\frac{a+b}{2c+3}$	<code>(a + b)/(2*c + 3)</code>
$\frac{a-c}{4bc}$	<code>(a - c)/(4*b*c)</code>

Suma de enteros

```
#include <iostream>
using namespace std;

int main()
{
    int entero1;           // primer numero que introduce el usuario
    int entero2;           // segundo numero que introduce el usuario
    int suma;              // variable en la que se almacena la suma
    cout << "Digite el primer entero\n";        // indicador
    cin >> entero1;         // lee un entero
    cout << "Digite el segundo entero\n";        // indicador
    cin >> entero2;         // lee un entero
    suma = entero1 + entero2;    // asignacion del resultado a suma
    cout << "La suma es " << suma << endl;      // impresion de suma
    return 0;
}
```

Ejercicio

Se desea un programa para calcular el área de un triángulo dadas las coordenadas de sus vértices.

- ▶ El área se calcula utilizando la siguiente fórmula:

$$\text{area} = \frac{1}{2}(x_1y_2 + x_2y_3 + x_3y_1 - x_1y_3 - x_2y_1 - x_3y_2)$$

- ▶ Variables de entrada: x_1 , x_2 , x_3 , y_1 , y_2 , y_3
- ▶ Variable de salida: area
- ▶ Casos de prueba:
 - entrada: (50, 25), (23, 30), (15, 15), salida: 222.5
 - entrada: (20, 15), (20, 0), (0, 0), salida: 150

Solución

```
#include <iostream>
using namespace std;

int main()
{
    float x1, x2, x3, y1, y2, y3; // coordenadas del triangulo
    float area;                  // area del triangulo

    cout << "Teclee las coordenadas del primer punto: ";
    cin >> x1 >> y1;
    cout << "Teclee las coordenadas del segundo punto: ";
    cin >> x2 >> y2;
    cout << "Teclee las coordenadas del tercer punto: ";
    cin >> x3 >> y3;

    area = (x1*y2 + x2*y3 + x3*y1 - x1*y3 - x2*y1 - x3*y2)/2.0;

    cout << "area: " << area << endl;
    return 0;
}
```

Funciones matemáticas

La biblioteca `math.h` contiene la definición de muchas funciones matemáticas útiles. El siguiente es un listado breve de algunas funciones.

Notación en C++	Función
<code>sin(x)</code>	seno de x
<code>cos(x)</code>	coseno de x
<code>tan(x)</code>	tangente de x
<code>sqrt(x)</code>	raíz cuadrada de x
<code>pow(x,y)</code>	x^y
<code>exp(x)</code>	e^x
<code>fabs(x)</code>	valor absoluto de x
<code>log(x)</code>	logaritmo natural de x
<code>log10(x)</code>	logaritmo base 10 de x
<code>ceil(x)</code>	entrega el menor entero que es mayor que x
<code>floor(x)</code>	entrega el mayor entero que es menor que x

Ejemplo

```
#include <iostream>
#include <math.h>
using namespace std;

int main()
{
    float a,b,c;  // lados del triangulo
    float s;      // semiperimetro
    float area;   // area del triangulo

    cout << "Teclee la longitud de los lados: ";
    cin >> a >> b >> c;  // lee los lados

    s = (a + b + c)/2.0;
    area = sqrt(s*(s-a)*(s-b)*(s-c));

    cout << "area: " << area << endl;
    return 0;
}
```
