

Programación en C++

1- Estructura general del un programa en C++

Un programa en c++ se compone de una o más funciones. Una de las funciones debe ser obligatoriamente **main**. Una función en C++ es un grupo de instrucciones que realizan una o más acciones. Asimismo un programa contendrá una serie de **#include** que permiten incluir archivos de cabecera que a su vez constarán de funciones y datos predefinidos.

Bloques de un programa C

Directivas del preprocesador

```
#include  
#define
```

Declaraciones globales

```
-- funciones  
-- variables
```

Función principal main

```
main()  
{  
  
}
```

Definiciones de otras funciones

```
funcion a ()  
{  
  
}
```

```
funcion b ()  
{  
  
}
```

2- Directivas del preprocesador

#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, Estos archivos se denominan archivos de cabecera o archivos de inclusión.

Los archivos de cabecera con extensión .h o .hpp contienen código fuente en (C / C++)
`#include <nombrearchivo.h>` o `#include "nombrearchivo.h"`

Los archivos más usuales de cabecera que se suele utilizar son `Iostream.H`. Este archivo proporciona al compilador c++ la información necesaria sobre las funciones de biblioteca `cin` y `cout`

Para la utilización de funciones de tratamiento de cadenas es necesario la incorporación del archivo de cabecera `string.h`. Para conservar la compatibilidad con funciones de entrada / salida de C, será preciso considerar la inclusión del archivo de cabecera `STDIO.H` mediante `include`.

Ejemplos de inclusión de código

- 1- `#include <iostream.h>` archivos que se encuentran en el directorio por defecto `include`
- 2- `#include "fichero.h"` archivos que se encuentran en el directorio actual

Los dos métodos no son excluyentes y pueden convivir en nuestro código

#define indica al preprocesador que defina un ítem de datos para sustitución del valor
ejemplo `#define Numero 45` sustituye el valor 45 cada vez que `Numero` aparezca en el programa

3- Función main

```
main
{
.... bloque de sentencias
}
```

* Una función en C++ es un bloque de código que devuelve un valor o un conjunto de valores o una tarea de E/S

*La función `main` constituye el punto de partida del programa

4- Comentarios

Sirven para introducir notas referidas al código o al algoritmo desarrollado por parte del programador no forman parte del código implementado por el compilador

Comentarios de una sola línea se utilizan los caracteres `//`

Comentarios de múltiples líneas se utilizan los caracteres `/* */`

ejemplos

```
// comentario esta línea no se implementa
```

```
/*  
    comienzo de un comentario  
    multilínea  
*/
```

5- Funciones

Las funciones de biblioteca son funciones listas para ejecutar que vienen con el c++, es necesario para su utilización la inclusión del archivo de cabecera.

Las funciones definidas por el programador debe ser declaradas previamente en el caso de no hacerlo así, se produce un error de compilación.

C++ es un lenguaje Tipado, en el caso de las funciones implica tener que declarar el tipo de dato devuelto de la función

ejemplo

```
int duplicar (int a); //declaración de la función duplicar
```

```
int duplicar ( int a ) //cuerpo de la función duplicar  
{  
    return a * 2;  
}
```

todas las sentencias terminan en punto y coma “;”, la cabecera del cuerpo de la función no

Nota-) Cuando una función no devuelve nada utilizamos la palabra reservada para el lenguaje

C++ void
ejemplo

```
void holamundo();
```

```
main ()  
{  
    holamundo(); //llamada a la función hola mundo  
}
```

```
void holamundo()  
{  
    cout << “hola mundo”;  
}
```

Tipo	Descripcion	Modificadores
------	-------------	---------------

void	Vacio	
------	-------	--

char	Caracter (8 bits)	signed char(8 bits),
------	-------------------	----------------------

		unsigned char(8 bits)
--	--	-----------------------

int Entero simple (16 bits) signed int(16 bits),
unsigned int(16 bits), long
int (32 bits), unsigned long
int(32 bits), signed long
int(32 bits), short int(16
bits), unsigned short
int(16 bits), signed short
int(16 bit)

float Coma flotante (32 bits)

double Coma flotante mas grande long double (80 bits)
(64 bits)

bool Valor booleano: true o
false

wchar_t Caracteres anchos, para
determinado juegos de
caracteres

6- Compilación

para instalación en linux sudo

para compilación g++ -o fichero.out fichero.cpp

para la ejecución ./fichero.out

7- Control de flujo

```
if ( condición a evaluar ) verdadera  
{  
    intrucciones a realizar  
}  
else  
{ instrucción a realiari en el caso de condición falsa
```

```
}
```

8- Ejercicios

primero.cpp

para compilar

```
g++ -o primero.out primero.cpp
```

para ejecutar

```
#include <iostream>
```

```
using namespace std;
```

```
main ()
```

```
{
```

```
cout << "hola mundo\n";
```

```
return 0;
```

```
}
```

segundo.cpp

```
using namespace std;
```

```
#include <iostream>
```

```
int main()
```

```
{
```

```
int x = 5;
```

```
int y = 6;
```

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

```
cout << "la suma de " << " " << x << " + " << y << " es igual a : " << x+y << "\n";
```

```
return 0;

}
tercero.cpp

/*
programa para la utilización de funciones
*/

using namespace std;
#include <iostream>

void funcion (int x, int y);

void funcion(int x,int y)
{
    cout << "la suma de x e y dentro de la funcion es " << x+y << "\n";
}

int main ()
{
    int x =5;
    int y =6;

    funcion(5,6);
```

```
return 0;
```

```
}
```

cuarto.cpp

```
/* función que devuelve un valor y se imprime desde el método main */
```

```
using namespace std;
```

```
#include <iostream>
```

```
int funcion (int x, int y);
```

```
int funcion (int x,int y)
```

```
{
```

```
return (x +y);
```

```
}
```

```
int main ()
```

```
{
```

```
int x = 5;
```

```
int y = 6;
```

```
int resultado = funcion(x,y);
```

```
cout << "el resultado de sumar" << x << " + " << y << " es " << resultado << endl;
```

```
cout << "el resultado de sumar imprimiendo el valor de la funcion es " << funcion (x,y) << endl;
```

```
return 0;
```

```
}
```

quinto.cpp

```
using namespace std;
```

```
#include <iostream>
```

```
int sumar ( int x , int y );
```

```
int sumar (int x, int y )
```

```
{
```

```
cout << " Estamos dentro de la función sumar " << endl;
```

```
return x + y;
```

```
}
```

```
int main ()
```

```
{
```

```
int x , y , z ;
```

```
cout << "Estamos dentro del main " << endl;
```

```
cout << "Introduce el primer sumando " << endl;
```

```
cin >> x;
```

```
cout << "Introduce el segundo sumando " << endl;
```

```
cin >> y;

z = sumar(x,y);

cout << "El resultado de sumar los dos sumandos introducidos es: " << z << endl;

return 0;
}
```

sexto.cpp

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

int main()
{
    cout << "El tamaño del int es:\t"<<sizeof(int)<<"bytes.\n";
    cout << "El tamaño del short es:\t"<<sizeof(short)<<"bytes.\n";
    cout << "El tamaño del long es:\t"<< sizeof(long)<<"bytes.\n";
    cout << "El tamaño del char es:\t"<< sizeof(char)<<"bytes.\n";
    cout << "El tamaño del float es:\t"<<sizeof(float)<<"bytes.\n";
    cout << "El tamaño del double es:"<<sizeof(double)<<"bytes.\n";

    return 0;

}
```

septimo.cpp

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

#define valor 25

int main ()
{
    int x=0;
    int y=0;

    const int Constante = 10;

    cout << "el contenido de valor es :" << valor << endl;

    cout << " el contenido de constante es :" << Constante << endl;

    return 0;
}
```

octavo.cpp

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

int variable_global = 10;

void mostrarFuncion ( );
```

```
void mostrarCaracteres ();
```

```
void mostrarCadenas();
```

```
void mostrarFuncion ()
```

```
{
```

```
    cout << "El valor de la variable global dentro de la función es : " << variable_global << endl;
```

```
}
```

```
void mostrarCaracteres ()
```

```
{
```

```
    char caracter_a = 'a';
```

```
    char caracter_b = 'b';
```

```
    //código ascii de a es 97
```

```
    caracter_a -= 32;
```

```
    char caracter_c = caracter_b + 1;
```

```
    cout << "El valor de caracter a es " << caracter_a << " el valor de caracter b es " << caracter_b << " el valor de caracter c es " << caracter_c << endl;
```

```
}
```

```
void mostrarCadenas ()
```

```
{
```

```
    char cadena [] = "hola esto es una cadeana ";
```

```
    char otraCadena [] = "\n esto es otra cadena ";
```

```
cout << cadena << otraCadena << endl;
}

int main ()
{

// llamada a la función mostrarFunción
mostrarFuncion ();

// llamada a la función mostrarCaracteres
mostrarCaracteres();

//llamamos a la función mostrarCadenas
mostrarCadenas();

cout << "El valor de la variable global es " << variable_global << endl;

}
```

noveno.cpp

```
//programa para utilización de operadores aritméticos

using namespace std;
#include <iostream>

int main()
```

```
{  
    cout << " Vamos a mostrar por pantalla utilización de operadores aritméticos \n";  
  
    //declaración y asignación en una sola línea  
  
    int a =0 , b =0 , c =0;  
  
    float d;  
  
    a = a +1;  
  
    b += 1;  
  
    cout << "el valor de la variable a es : " << a << endl;  
    cout << "el valor de la variable b es : " << b << endl;  
  
    c = ++a;  
  
    cout << "el valor de la variable c es : " << c << endl;  
  
    b = c++;  
  
    cout << "el valor de la variable b es : " << b << "\nel valor de la variable c es: " << c << endl;  
  
    c *= 2;  
  
    cout << "el valor de la variable c es el doble " << c << endl;  
  
    c %= 2;
```

```
cout << "el valor de la variable c es el módulo de su valor entre dos " << c << endl;
```

```
cout << "introduce un numero por teclado " << endl;
```

```
cin >> a;
```

```
if (a >= 0)
```

```
{
```

```
    cout << "la variable a tiene un valor positivo " << endl;
```

```
}
```

```
else
```

```
{
```

```
    cout << "la variable a tiene un valor negativo " << endl;
```

```
}
```

```
cout << "introduce por teclado un numero " << endl;
```

```
cin >> b;
```

```
if ( a == b)
```

```
{
```

```
    cout << "los dos numeros son iguales a y b " << endl;
```

```
}
```

```
cout << "introduce por teclado un numero " << endl;
```

```
cin >> d;
```

```
if ( b == d )  
{  
    cout << "tienen el mismo valor " << endl;  
}
```

```
if ( b != d )  
{  
    cout << "tienen distinto valor " << endl;  
}
```

```
return 0;
```

```
}
```