



# Primeros pasos

# Índice

- Esqueleto de un programa.
- Nombre de variables.
- Concepto casting.
- Flujos de entrada o salida.
- Salida de datos.
- Entrada de datos.
- Funciones
- Ejercicios
- Ejercicios entregables



# **ESQUELETO DE UN PROGRAMA**

# Esqueleto de un programa

- El código fuente de un programa se estructura de la siguiente manera:
  - Declaración de librerías.
  - Uso del espacio de nombres std.
  - Declaración de constantes.
  - Declaración de variables globales y funciones.
  - Función principal: `int main()`
    - Código del programa
  - Definición de las funciones.

# Esqueleto de un programa

- Ejemplo de estructura básica:

```
#include <iostream>
```

```
#include "stdlib.h"
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
//Instrucciones de codigo
```

```
}
```



# **NOMBRE DE VARIABLES**

# Nombre de variables

- Sólo pueden utilizarse caracteres alfanuméricos estándar (no Ñ o acentos) y el carácter subrayado ‘\_’ No puede utilizarse el carácter blanco.
- El primer carácter ha de ser una letra o el carácter subrayado.
- No pueden utilizarse palabras reservadas como nombres de variables o funciones.
- El C++ se distingue entre mayúsculas y minúsculas, por lo que hay que tener cuidado de cómo se escribe un identificador.

Válidos	No válidos
num2	año
PracticaDos	nombre usuario
_uno	2num
precio_compra	Matricula_Camión

# Nombre de variables

Palabras reservadas

- Palabras que tienen un significado especial en el lenguaje y no pueden ser utilizadas para ninguna otra cosa. Definen la estructura del programa y las instrucciones más básicas.

ejemplo





# **CONCEPTO CASTING**

# Concepto casting

- Conversión implícita de datos
  - Cuando se evalúa una expresión algebraica que incluye distintos tipos de datos. El resultado de dicha operación será de tipo de dato igual al de mayor capacidad en bits:
    - $5,0 + 2 = 7,0$  (el dato float tiene mayor capacidad que el int)
- Conversión explícita de datos (casting)
  - Se transforma el tipo de dato a uno diferente al que estamos trabajando .
    - $\text{char}(65) = 'A'$                        $\text{int}(5,7) = 5$
    - $\text{int}('A') = 65$                            $\text{float}(5) = 5,0$



# **FLUJOS DE ENTRADA O SALIDA**

# Flujos de entrada o salida

- Entrada de datos
  - Normalmente por teclado.
- Salida de datos
  - Normalmente por la pantalla.



# **SALIDA DE DATOS**

# Salida de datos

- Podemos enviar por el flujo de salida cualquier combinación de variables y cadenas de texto por pantalla:
- Las instrucciones más usadas son:
  - `printf`
  - `cout`

# Salida de datos printf

```
nombre="Miguel";  
edad=18;
```

```
printf("Mi nombre es %s y tengo %i años", nombre, edad);
```


# Salida de datos printf

Parametro	Significado
%d o %i	Entero con signo
%u	Entero sin signo
%o	Octal sin signo
%x	Entero hexadecimal con signo
%X	Entero hexadecimal sin signo
%f	Decimal, coma flotante, corto
%F	Decimal, coma flotante, largo
%e	Notacion cientifica (mantissa/exponente), 32 bits
%c	Carácter
%s	Cadena de caracteres
%P	Direccion de memoria (puntero)



# Salida de datos cout

```
cout << num_dulces;  
cout << "dulces";
```



```
cout << num_dulces << "dulces";
```

Se pueden incluir expresiones aritméticas

```
cout << "El precio total es:" << (precio1+precio2);
```

Caracteres especiales	
\n	Nueva línea
\t	Tabulación horizontal
\\	Diagonal invertida
\"	Comillas dobles

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



```
cout << endl;
```

Funciones interesantes de cout



# **ENTRADA DE DATOS**

# Entrada de datos

- Podemos leer por el flujo de entrada información del teclado de un tipo de datos concreto: enteros, decimales, letras, palabras...
- Las instrucciones más usadas son:
  - `scanf`
  - `cin`
  - `getch` / `getche`

# Entrada de datos scanf

- El tipo de parámetro es el mismo que en printf.

```
int numero;
```

```
printf("Introduce tu edad:");
```

```
scanf("%d", &numero);
```

# Entrada de datos cin

```
int numero;  
printf("Introduce tu edad:");  
cin>>numero;
```

# Entrada de datos cin

```
#include "iostream"
#include "string"
using namespace std;
int main()
{
    int numero1, numero2;
    cout << "Hola! Aqui podras realizar sumas" << endl;
    cout << "Por favor ingrese el primer valor: " << endl;
    cin >> numero1;
    cout << "Por favor ingrese el segundo valor: " << endl;
    cin >> numero2;
    cout << "El resultado de la suma de " << numero1 << " + " <<
    numero2 << " es: " << numero1+numero2 << endl;
    return 0;
}
```

# Entrada de datos *getch/getche*

Si lo que queremos es que el usuario introduzca un carácter por el teclado usamos las funciones *getch* y *getche*. Estas esperan a que el usuario introduzca un carácter por el teclado.

La diferencia entre *getche* y *getch* es que la primera saca por pantalla la tecla que hemos pulsado y la segunda no (la *e* del final se refiere a *echo=eco*).

# Entrada de datos getch/getche

```
#include <stdio.h>
#include <conio.h>
int main()
{
    char letra;
    printf( "Introduce una letra: " );
    fflush( stdout );
    letra = getche();
    printf( "\nHas introducido la letra: %c", letra );
}
```





# **FUNCIONES**

# Funciones

- Las **funciones** son un conjunto de instrucciones que realizan una tarea específica.
- En general toman ciertos valores de entrada, llamados parámetros y proporcionan un valor de salida o valor de retorno.
- Tanto unos como el otros son opcionales, y pueden no existir.

# Funciones

- Se declaran antes de la función principal main.
- Se llaman dentro de la función main o puede que dentro de otra función.
- Se definen fuera de la función main, por lo general a continuación.
- Existen las funciones simples y las recursivas, las externas y estáticas.

# Funciones declaración

Tipo\_de\_dato\_que\_retorna Nombre (argumentos);

Para indicar que no va existir un valor de retorno se utiliza la palabra void, mientras que si no existen argumentos los paréntesis quedarán vacíos.

Ejemplos:

```
void Saludo();
```

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

```
float Area_Circunferencia(float radio);
```

# Funciones definición

```
int suma(int x, int y)
{
    int suma;
    suma=x+y;
    return suma;
}
```

# Funciones llamada

```
int a ,b, resultado;
```

```
a=10;
```

```
a=8;
```

```
resultado=suma(a,b);
```

o

```
resultado=suma(10,8);
```

# Funciones paso por referencia

```
int funcion(int n, int m);  
int main()  
{  
  int a, b;  
  a = 10;  
  b = 20;  
  cout << "a,b ->" << a << ", " << b << endl;  
  cout << "funcion(a,b) ->" << funcion(a, b) << endl;  
  cout << "a,b ->" << a << ", " << b << endl;  
  return 0;  
}  
int funcion(int n, int m)  
{  
  n = n + 2;  
  m = m - 5;  
  return n+m;  
}
```

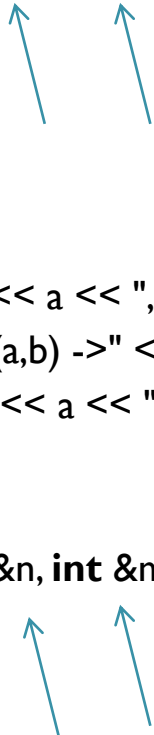
# Funciones paso por referencia

- Si queremos que los cambios realizados en los parámetros dentro de la función se conserven, deberemos pasarlos por referencia.



# Funciones paso por referencia

```
int funcion(int &n, int &m);  
int main()  
{  
    int a, b;  
    a = 10;  
    b = 20;  
    cout << "a,b ->" << a << ", " << b << endl;  
    cout << "funcion(a,b) ->" << funcion(a, b) << endl;  
    cout << "a,b ->" << a << ", " << b << endl;  
    return 0;  
}  
int funcion(int &n, int &m)  
{  
    n = n + 2;  
    m = m - 5;  
    return n+m;  
}
```





# **EJERCICIOS**

# Ejercicios

- Programa que sume 2 números pedidos al usuario.
- Programa que pida dos números y calcule su media.
- *Programa que calcula la división y el resto de dos números introducidos por el usuario*
- Realizar un programa que pida dos números al usuario, luego los intercambie y los muestre otra vez por pantalla.

# Ejercicios

- Realice un programa que pida al usuario:
  - su nombre
  - sus apellidos
  - los kilómetros existentes de su casa al trabajo.

Y muestre por pantalla una nomina sencilla que incluya:

- su salario base(956,28 €)
- su plus por transporte ( $\text{km} \times 0,85$ )
- las contingencias comunes (4,7%)
- el desempleo (1,55%)
- la formación (0,10%)
- el total de aportaciones
- el IRPF que esta cifrado en 11%
- el salario neto total



# **EJERCICIOS ENTREGABLES**

# Ejercicios entregables (1/2)

- Escriba un programa que muestre por pantalla el mensaje de Hola.
- Realizar un programa que pida al usuario un nombre y la edad y muestre su edad y nombre por la pantalla.
- Realice un programa que pida al usuario una letra y escriba por pantalla “La letra que usted ha tecleado es ” seguido de la letra introducida.
- Realice un programa que pida una temperatura en grados centígrados, los convierta en grados kelvin y muestre el resultado [sugerencia: para conseguir la conversión de grados centígrados a kelvin, hay que sumar al valor de grados centígrados 274,15]
- Realice un programa que pida dos números al usuario, realice una división y muestre el resultado.

# Ejercicios entregables (2/2)

- Programa que pida las notas de tres asignaturas: física, geografía y educación física y muestre la nota media.
- Programa que calcule la división y el resto de dos números introducidos por el usuario.
- Programa que una vez introducido el precio neto de un producto muestre por pantalla el iva (21%) y el precio total.
- Un programa utiliza unas donaciones de ONG. Estas donaciones han sido abonadas, y tienen que ser repartidas de la siguiente manera:
  - 60% para un centro de salud.
  - 35% para un comedor para niños.
  - El resto para gastos administrativos.

*Muestra por pantalla la cantidad destinada a cada partida. (este ejercicio vale doble)*