

# *Informática II*

## Sesión 05: Fundamentos C++

# *Contenido*



**Introducción a C++**



**Variables**



**Operadores**



**Estructuras de control**

# *Contenido*



**Introducción a C++**



**Variables**



**Operadores**



**Estructuras de control**

# Introducción

Los lenguajes de programación han sufrido una dramática evolución desde los primeros computadores.

□ Lenguaje de maquina.

```
1010010010011  
1110101010100  
1010100101111  
000111
```

# Introducción

- Lenguaje de programación de Bajo nivel.
- Ventajas:
  - El programa optimiza el uso de los recurso de la maquina.
  - Programas veloces en su ejecución.
- Desventajas:
  - Dependientes de la maquina. (No portables)
  - Necesidad de conocimientos avanzados tanto de programación como de la maquina.

**Assembler (Ensamblador)**  
**Cada arquitectura tiene el propio.**

```
MOV AX, @data;  
MOV DS, AX;  
MOV DX, offset;  
MOV AH, 9;  
INT 21h;
```

# *Introducción*

- Sin embargo, las tareas realizadas por las aplicaciones de software eran cada vez más complejas, por ejemplo: las trayectorias de los misiles.
- Necesitaban un lenguaje que les permitiera realizar instrucciones matemáticas complejas (lenguaje ensamblador eran muchas instrucciones).

# Introducción

- Lenguaje de programación de Alto nivel.
- Ventajas:
  - Se basan en reglas similares a los lenguajes humanos
  - Modificables fácilmente.
  - Independiente de la máquina. Son Portables.
- Desventajas:
  - Herramientas complejas de traducción a L.M.
  - No se optimizan los recursos de la maquina.
  - Programas mas lentos en su ejecución

**C, C++, C#, FORTRAN, Pascal,  
Visual BASIC, Java.  
HTML, XML, JavaScript**

```
#include <iostream.h>
int main (void)
{
    char x;
    cout <<"Introduce un carácter"<<endl;
    cin <<a;
    if(a>="A" && a<="a" || a>=48 && a<=57)
        cout <<"
    }
}
```

# ***Introducción***

- El lenguaje C (creado por Dennis Ritchie) fue la evolución de un versión anterior llamada B, el cual era una mejora del lenguaje de BCPL (Basic Combined Programming Language).
- C es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix.



# Introducción

- Por su parte, C++ fue desarrollado por Bjarne Stroustrup en los laboratorios de bell (St86), originalmente fue llamado “C con clases”. Se cambió usando el operador de incremento (++), para indicar que es una versión mejorada de C
- C++ es una mejora sobre muchas de las características de C, y proporciona capacidad de programación orientada a objetos. (Productividad, calidad, reutilización).

# Introducción

## ● Intérpretes

- Leen código fuente tal y como está escrito y lo van ejecutando.
- Puede ser más fácil para el programador.
- Scripts – Script Engines.

## ● Compiladores

- Traduce código fuente a una forma intermedia (compilado).
- El enlazador crea el archivo ejecutable (en lenguaje de máquina).
- Puede correr directamente sobre el procesador.

# *Introducción*

- Interpretados

- Interpretación en tiempo de ejecución.
- Requiere más tiempo para ejecutarse.
- Intérprete debe estar instalado en la máquina.

- Compilados

- Compilación off-line.
- Distribuible.

# ***Introducción***

- Estructura de un programa
- Variables
- Operadores
- Entrada y salida
- Estructuras de control
- Funciones
- Apuntadores

# *Contenido*

1

**Introducción a C++**

2

**Variables**

3

**Operadores**

4

**Estructuras de control**

# ***Variables***

# *Variables*

- Una variable es un lugar en memoria para almacenar información.
- Cada variable tiene un tamaño específico que le dice a la máquina cuanta memoria necesita reservar.
- Existen variables locales y globales.
- Case sensitive.

# Variables

## ● Tipos de datos

- En C++ es necesario definir el tipo de cada variable que se va a usar. El tipo de variable define el espacio de memoria que esta ocupa y la forma como se interpreta la información contenida en ella.

Nombre	Bytes	Rango
char	1	-127 a 127
int	4	-2147483648 a 2147483647
float	4	+/-3.4E+/-38 (7 digitos)
double	8	+/-1.7E+/-308(15 digits)
bool	1	true o false
void	0	Sin valor
wchar_t	2	0 a 65,535



# Variables

## ● Tipos de datos

- En C++ es necesario definir el tipo de cada variable que se va a usar. El tipo de variable define el espacio de memoria que esta ocupa y la forma como se interpreta la información contenida en ella.

Nombre	Bytes	Rango
char	1	-127 a 127
int	4	-2147483648 a 2147483647
float	4	+/-3.4E+/-38 (7 digitos)
double	8	+/-1.7E+/-308(15 digits)
bool	1	true o false
void	0	Sin valor
wchar_t	2	0 a 65,535

# Variables

- Modificadores de signo

- C++ trata por defecto a todas las variables como datos con signo. Es posible utilizar los modificadores *signed* y *unsigned* para modificar como son interpretados estos datos y, por lo tanto, modificar su rango.

Nombre	Bytes	Rango
signed char	1	-127 a 127
unsigned char	1	0 a 255
signed int	4	-2147483648 a 2147483647
unsigned int	4	0 a 4,294,967,295

# Variables

## ● Modificadores de tamaño

- El tamaño de una variable tipo *int* puede ser aumentado o disminuido usando los modificadores *short* y *long*, con estos modificadores se obtienen los siguientes tipos de variables:

Nombre	Bytes	Rango
short int	2	-32,768 a 32,767
long int	4	0 a -2147483648 a 2147483647
long long int	8	-9,223,372,036,854,775,808 a 9,223,372,036,854,775,807

- ¿*long double* existe?



# *Variables*

- Declaración
  - Modificador de signo + modificador de tamaño + tipo + nombre.

# Variables

- Declaración

- Modificador de signo + modificador de tamaño + tipo + nombre.

- Ejemplos

`int i;`

# Variables

- Declaración

- Modificador de signo + modificador de tamaño + tipo + nombre.

- Ejemplos

`int i;`

`signed int i;`

# Variables

- Declaración

- Modificador de signo + modificador de tamaño + tipo + nombre.

- Ejemplos

`int i;`

`signed int i;`

`unsigned int i;`

# Variables

- Declaración

- Modificador de signo + modificador de tamaño + tipo + nombre.

- Ejemplos

`int i;`

`signed int i;`

`unsigned int i;`

`long int i;`



# Variables

- Declaración

- Modificador de signo + modificador de tamaño + tipo + nombre.

- Ejemplos

`int i;`

`signed int i;`

`unsigned int i;`

`long int i;`

`unsigned long int i;`

# Variables

## ● Declaración

```
int a;
```

```
float mynumber;
```

```
int a, b, c;
```

```
int a;
```

```
int b;
```

```
int c;
```

# Variables

## ● Inicialización

Hay tres maneras de inicializar variables:

```
int x = 0;
```

```
int x (0);
```

```
int x {0}
```

# Bibliografía

## Libros:

- B. Eckel. *Pensar en C++*. 2000.  
[http://arco.esi.uclm.es/~david.villa/pensar\\_en\\_C++/vol1/](http://arco.esi.uclm.es/~david.villa/pensar_en_C++/vol1/)
- H. Deitel y P. Deitel, *Como programar en C/C++*, 6 Ed., Pearson Education, 2009.
- J. Liberty, S. Rao, B. Jones . *Sams Teach Yourself C++ in One Hour a Day*.

## Internet:

- Google
- [www.cplusplus.com](http://www.cplusplus.com)