



# El lenguaje C++



**B.S. Rodolfo Mercado Gonzales**

# ¿Qué es C++?

- ❑ Lenguaje de programación compilado y de “alto nivel”, creado en 1980 por Bjarne Stroustrup.
- ❑ Es una mejora del lenguaje C, la principal funcionalidad agregada es el manejo de objetos.

# Versiones Estándar de C++

- ❑ **C++98**, incluye la STL, ...
- ❑ **C++11**, deduce tipo de dato, hash table, expresiones regulares, ...
- ❑ **C++14**, deduce tipo de dato para retorno de funciones, ...
- ❑ **C++17**, ...

# Programa Básico en C++

```
1 #include <iostream>
2
3 int main()
4 {
5     std::cout << "Hola mundo";
6     return 0;
7 }
```

**Línea 1 :** encabezados, permiten el uso de ciertas funcionalidades.

**Línea 2 :** líneas en blanco no afectan.

**Línea 3 :** función principal, es llamada al iniciar un programa.

**Líneas 4 y 7 :** inicio y fin de una función.

**Líneas 5 y 6 :** instrucciones en C++.

# Ambiente de Programación – Opción 1

## Editor de Texto

gedit, geany, sublime, atom, ...



## Compilador

GNU Compiler Collection : g++

\* en Windows usar TDM-GCC o Mingw-w64



# Ambiente de Programación – Opción 2

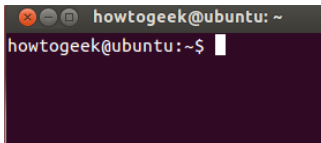
## IDE (Integrated Developmet Enviroment)

- ❑ Un entorno de desarrollo integrado proporciona funcionalidades que facilitan la tarea de programación.
- ❑ Generalmente incluyen editor de texto, compilador, depurador, autocompletado, ...
- ❑ Code::Blocks, Dev C++, CLion, eclipse, ...



# Compilación y Ejecución

Necesitamos usar la terminal (cmd)



**Código Fuente**  
fuente.cpp



**Ejecutable**  
a.exe, a.out,...

**compilación**  
g++ fuente.cpp

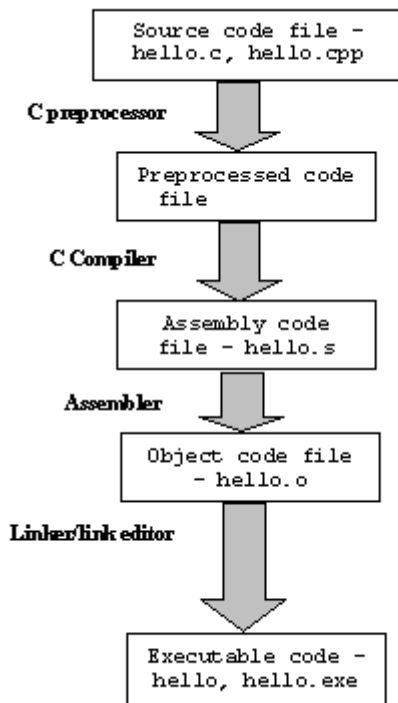
**ejecución**  
a.exe ./a.out

# Proceso de Compilación

Los compiladores C++ usan wrappers que automatizan todo el proceso.

Para detener la compilación después de alguna fase, usar:

- **-E**, preprocessed code
- **-S**, assembly code
- **-c**, object code





# Función main

- ❑ Es la primera función a ser llamada, el resto de funciones son llamadas directa o indirectamente por ésta.
- ❑ El retornar **0** indica que el programa terminó con éxito.

```
int main() {  
}
```

```
int main( int argc, char* argv[] ){  
}
```

# Preprocesador

- ❑ Procesa directivas
- ❑ Utiliza una sintaxis diferente a C++.
- ❑ Las directivas inician con el símbolo **#** y terminan con el final de la línea.
- ❑ Las directivas más comunes son **#include** y **#define**.

# Preprocesador / #include

- ❑ Directiva que permite al programa usar código fuente de otro archivo.
- ❑ Por ejemplo **#include <iostream>** le dice al preprocesador que inserte el archivo iostream dentro del programa actual.
- ❑ Los archivos que se incluyen en otros programas se llaman archivos de encabezado.
- ❑ Para usar cin y cout necesitamos incluir el encabezado **iostream**.

# Variables y Constantes

```
char letra;  
string nombre;  
bool flag;  
int edad;  
long long nroTransacciones;  
float costo;  
double area;
```

```
const int MESES = 12;  
const int DIAS_SEMANA = 7;  
const double PI = acos(-1.0);
```

<b>int</b>	[-2147483648, 2147483647]
<b>long long</b>	[-9,223,372,036,854,775,808, 9,223,372,036,854,775,807]

# Lectura y Escritura

- ❑ La entrada estándar viene desde teclado.
- ❑ La salida estándar se muestra en consola
- ❑ Ambos pueden ser redireccionados mediante línea de comandos.

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

int main(){
    int a, b, c;
    cin >> a >> b; //lectura
    c = a + b;
    cout << c << endl; //escritura
}
```

# Lectura y Escritura

- ❑ **scanf** y **printf** nos permiten realizar lectura y escritura con formato.
- ❑ Proviene del lenguaje C.

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    char nombre[ 10 ];
    double nota1, nota2;
    scanf("%s", nombre);
    scanf("%lf%lf", &nota1, &nota2);
    printf("Promedio de notas de %s es %.2lf", nombre, (nota1 + nota2) / 2 );
}
```

<b>%d</b>	int
<b>%lld</b>	long long
<b>%f</b>	float
<b>%lf</b>	double
<b>%c</b>	char
<b>%s</b>	char[ ]

# Namespace

- ❑ Para organizar gran cantidad de variables y/o funciones, así como para evitar homonimia en éstas, en C++ se define el *espacio de nombres* o **namespace**.
- ❑ Todas las variables declaradas dentro de un **namespace** son consideradas de la misma familia.

```
#include <iostream>

namespace uno{
    void imprimir(){
        std::cout << "uno";
    }
}

namespace dos{
    void imprimir(){
        std::cout << "dos";
    }
}

int main(){
    uno::imprimir();
}
```

# Namespace std

- ❑ C++ define sus objetos y funciones estándar en el namespace std.

```
iostream
namespace std{
    istream cin;
    ostream cout;
    ...
}
```

Para acceder a alguna de éstas deberíamos colocar por ejemplo: **std::cin** , pero si queremos evitar escribir todo, sólo basta colocar **using namespace std** al inicio del programa y ahora solo necesitamos escribir **cin**.



# Operadores

La mayoría de operadores son iguales a como lo representábamos en pseudocódigo a excepción de:

Operador	Significado
%	Resto o módulo
/	División entera si ambos términos son enteros, en otro caso devuelve un real
!	negación
&&	conjunción
	Disyunción

# Estructura Selectiva / If Else

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

int main() {
    cin >> miEdad >> tuEdad;
    if( miEdad < tuEdad ) {
        cout << "soy menor";
    }
    else {
        cout << "soy mayor o tenemos la misma edad";
    }
}
```

# Estructura Repetitiva / While

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

int main(){
    int num;
    //mostrando los números del 1 al 10
    while( num <= 10 ){
        cout << num << endl;
        num = num + 1;
    }
}
```

# Estructura Repetitiva / For

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

int main() {
    //mostrando los números del 1 al 10
    for( int i = 1; i <= 10; i++ ){
        cout << i << endl;
    }
}
```

# Arreglos

```
double notas[ 20 ];  
char grid[ 50 ][ 50 ];  
int edades[ 10 ];
```

```
char nombre[ 5 ] = {'L', 'u', 'a', 'n', '\0'};  
string nombre = "Luan";
```

```
char nombres[ 30 ][ 5 ];  
string nombres[ 30 ];
```

# Funciones y procedimientos

```
int factorial( int n ){  
    ret = 1;  
    for( int i = 1; i <= n; ++i ){  
        ret = ret * i;  
    }  
}
```

```
void printArray( int A[ ], int n ){  
    for( int i = 1; i < n; ++i ){  
        cout << " " << A[ i ];  
    }  
}
```

# Preprocesador / #define

- ❑ Permite definir las denominadas macros.
- ❑ La macros tienen el siguiente formato:

*#define nombreMacro reemplazo*

- ❑ El preprocesador reemplaza todas la ocurrencias de **nombreMacro** en el programa y los reemplaza por **reemplazo**, antes de iniciar la compilación.

```
#define pb push_back
#define FOR( i , n ) for( int i = 0 ; i < n ; i++ )
```

# Problemas

Codeforces – Game With Sticks

HackerRank - Compare the Triplets

HackerRank – Array Introduction

HackerRank – A Very Big Sum

HackerRank – Strings



# Referencias

- ❑ Oualline, Steve. Practical C++ programming.
- ❑ Stroustrup, Bjarne . The C++ Programming Language.
- ❑ Hackerearth. The Build Process – C/C++.

¡ Good luck and have fun !