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 de variables, for basado en rangos, mejora del pair, …
- **C++14, ...**
- **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

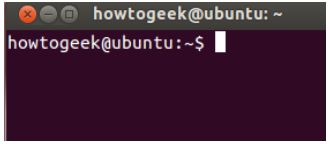
Ambiente de programación – opción 2

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

compilación
g++ fuente.cpp

Ejecutable
a.exe, a.out,…

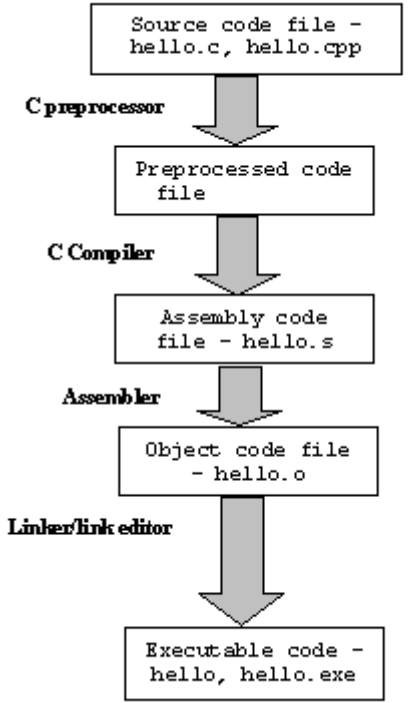
ejecución
a.exe ./a.out

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);
```

tipo de dato	valores
int	$[-2147483648, 2147483647]$
long long	$[-9{,}223{,}372{,}036{,}854{,}775{,}808,$ $9{,}223{,}372{,}036{,}854{,}775{,}807]$

Pair

- Agrupa un par de elementos, los cuales pueden ser de distintos tipos de datos.

```
pair<int, int> p1, p2;
p1.first = 3, p1.second = 5;
p2 = {-4, 6};
```

Lectura y escritura

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

```cpp
#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.
- Provienen del lenguaje C.

```
#include <cstdio>
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 );
}
```

%d	int
%lld	long long
%f	float
%lf	double
%c	char
%s	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 los representamos 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

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

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

Estructura Repetitiva / For basado en rango

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

int main() {
    int A[] = {5, 9, 100};
    for( int elem : A ){
        cout << elem << 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++ )
```

Auto

- Deduce el tipo de dato de una variable, a partir de la expresión usada para su inicialización.

```
auto cadena = "hola";
auto real = 5.24;

int A[] = {5, 10, 6};
for( auto elem : A ){
    cout << elem << endl;
}
```

Problemas

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 !