

01 Introducción

Programación Orientada a Objetos

Néstor Suat-Rojas. Ing. Msc (c)
nestor.suat@unillanos.edu.co

Escuela de Ingeniería
Facultad de Ciencias Básicas e Ingeniería

Introducción



Introducción



Lenguaje de Programación de propósito general muy poderoso y de alto nivel, incluyendo la capacidad de agregar código assembler.

Introducción



Es un lenguaje multiparadigma.
“C con clases” o “Incremento de C”.



**UNIVERSIDAD
DE LOS LLANOS®**

Introducción

Este lenguaje fue creado a principios de los noventa por Bjarne Stroustrup en los laboratorios de AT&T en 1979.

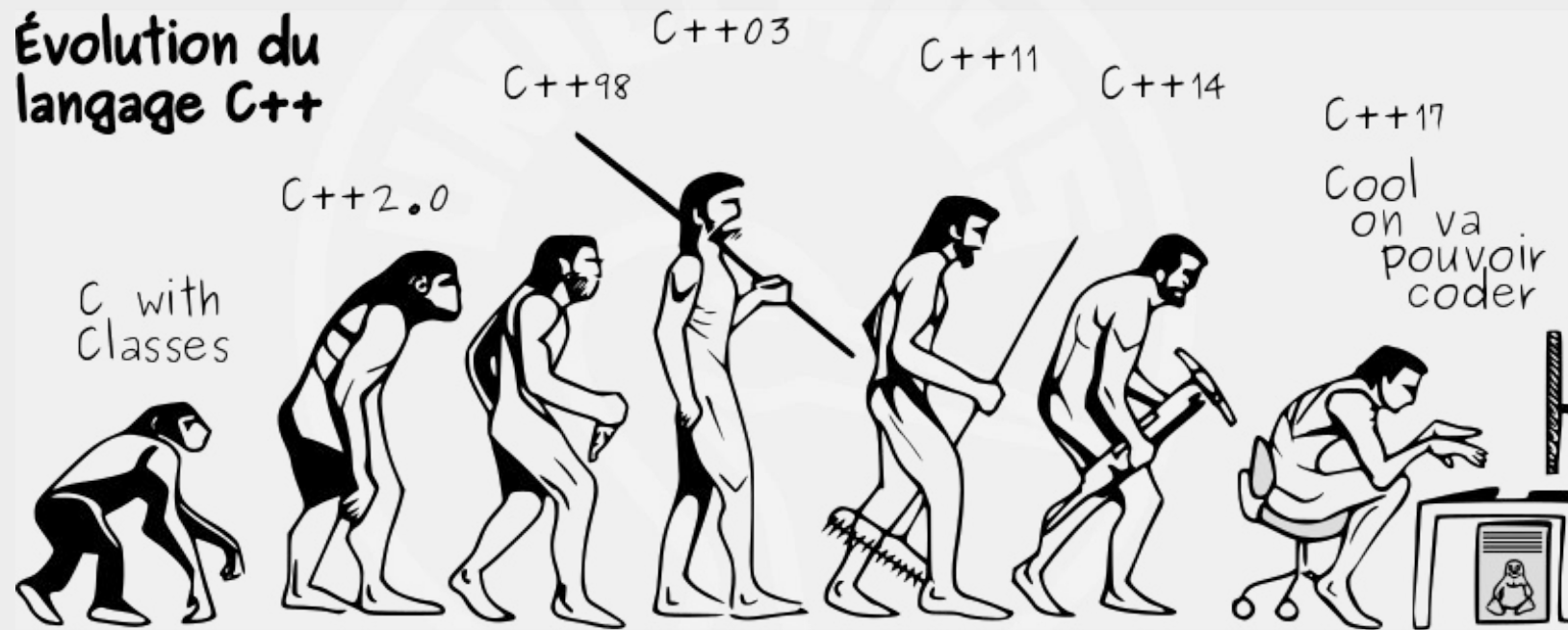
Es una alternativa madura de Fortran 1957, Pascal 1970 e incluso C 1972.

Bjarne Stroustrup



**UNIVERSIDAD
DE LOS LLANOS®**

Introducción

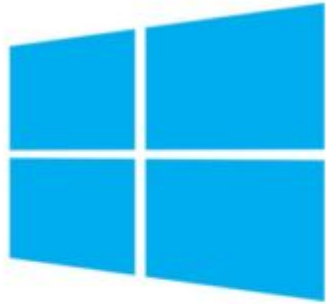


Estándar ISO/IEC C++20



**UNIVERSIDAD
DE LOS LLANOS®**

Introducción



Windows 10

- Visual C++
- Visual Basic
- C/C++ con Qt o GTK
 - Java con AWT, Swing o JavaFX



MacTMOS

- Objective-C/Swift con Cocoa
- C/C++ con Qt o GTK
 - Java con AWT, Swing o JavaFX



Linux

- C/C++ con Qt o GTK
- Java con AWT, Swing o JavaFX



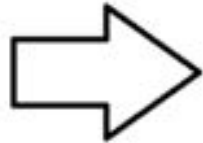
**UNIVERSIDAD
DE LOS LLANOS[®]**

Introducción

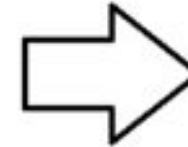
```
#include<stdio.h>

int main()
{
    printf("Hello, World!\n");
    return 0;
}
```

hello_world.c



Compiler



```
0110011000100010011000111
1100000001111111110000001
1111000110101010001100011
0011000100010011000111110
0000001111111110000001111
1000110101010001100011001
1000100010011000111110000
0001111111110000001111100
0110101010001100011001100
0100010011000111110000000
1111111110000001111100011
0110100011000110011000110
```

hello_world.o



UNIVERSIDAD
DE LOS LLANOS®

Configuración C++

- Descargar e instalar Compiler.
 - **GCC (g++) - MinGW-W64**
 - Glang
- Editor de texto o IDE.
 - Sublime Text
 - Visual Code
 - **CLion**

Sintaxis C++

1. C++ es un lenguaje sensible a las mayúsculas y minúscula.
 - Ej: NOMBRE y nombre no son los mismos que para el lenguaje C++
2. El nombre de la clase debe comenzar con mayúscula.
3. El nombre del método debe comenzar con minúscula.
4. Cada declaración debe terminar con punto y coma.
5. La ejecución de un programa C++ comienza desde el método principal, que es obligatorio en todos los programas.

```
int main() {  
    return 0;  
}
```

Imprimir en C++

Sintaxis:

```
std::cout << "Hola mundo";
```

Imprimir con salto de línea:

```
std::cout << "Hola mundo" << endl;
```

Leer variables por el usuario:

```
std::cin >> variable;
```

Comentarios en C++

Sintaxis:

```
1 // Single line comment
```

```
1 /* Multi line comments - Line 1  
2 Multi line comments - Line 2  
3 */
```



**UNIVERSIDAD
DE LOS LLANOS®**

Variables en C++

Sintaxis:

```
tipodato nombre_de_variable = valor;
```

```
int num1 = 0;
```

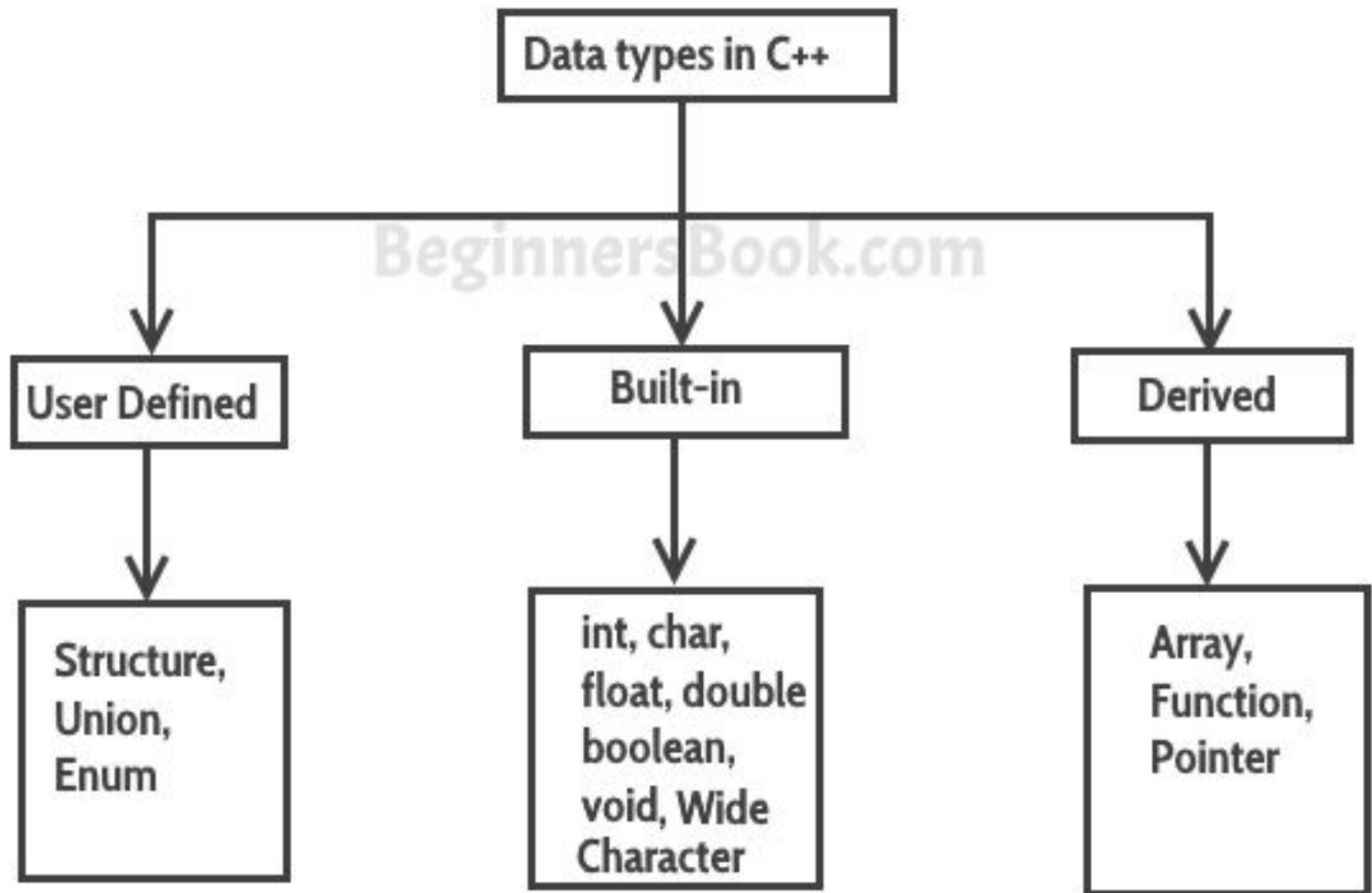
Usando caracteres especiales

```
int $num2= 0;
```

Usando mayúsculas

```
int Age = 0;
```

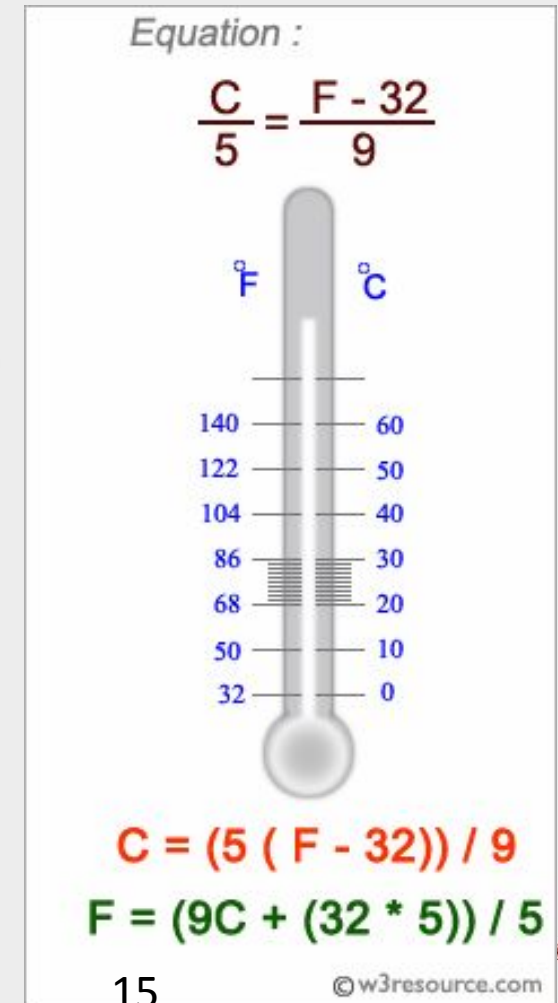
Variables en C++



Variables en C++

Ejercicios

1. Sumar dos números.
2. Multiplicar dos números.
3. Conversión de Fahrenheit.
 - a. De Fahrenheit a Celsius:
4. Halar índice de masa corporal.
 - a. Input: Peso y la Altura.



Operadores en C++

- Operadores aritméticos (+, −, *, /, %)
- Operadores de asignación (=, +=, -=, *=, /=, %=)
- Operadores de autoincremento y autodecremento (++ , −)
- Operadores lógicos (&&, ||, !)
- Operadores de comparación (==, !=, >, <, >=, <=)
- Operadores a nivel de bit(&, |, ^, ~, <<, >>)

Operadores

- Operadores de asignación (=, +=, -=, *=, /=, %=)

```
num2 = num2 + num1;
```

```
#include <iostream>
using namespace std;
int main(){
    int num1 = 240;
    int num2 = 40;
    num2 = num1;
    cout<<"= Output: "<<num2<<endl;
    num2 += num1;
    cout<<"+= Output: "<<num2<<endl;
    num2 -= num1;
    cout<<"-= Output: "<<num2<<endl;
    num2 *= num1;
    cout<<"*= Output: "<<num2<<endl;
    num2 /= num1;
    cout<<"/= Output: "<<num2<<endl;
    num2 %= num1;
    cout<<"%= Output: "<<num2<<endl;
    return 0;
}
```

Operadores en C++

- Operadores de autoincremento y autodecremento (++ , --)

```
#include <iostream>
using namespace std;
int main(){
    int num1 = 240;
    int num2 = 40;
    num1++; num2--;
    cout<<"num1++ is: "<<num1<<endl;
    cout<<"num2- - is: "<<num2;
    return 0;
}
```

Operadores en C++

- Operadores lógicos (&&, ||, !).

```
#include <iostream>
using namespace std;
int main(){
    bool b1 = true;
    bool b2 = false;
    cout<<"b1 && b2: "<<(b1&&b2)<<endl;
    cout<<"b1 || b2: "<<(b1||b2)<<endl;
    cout<<"!(b1 && b2): "<<!(b1&&b2);
    return 0;
}
```

Operadores en C++

- Operadores de comparación. (==, !=, >, <, >=, <=)

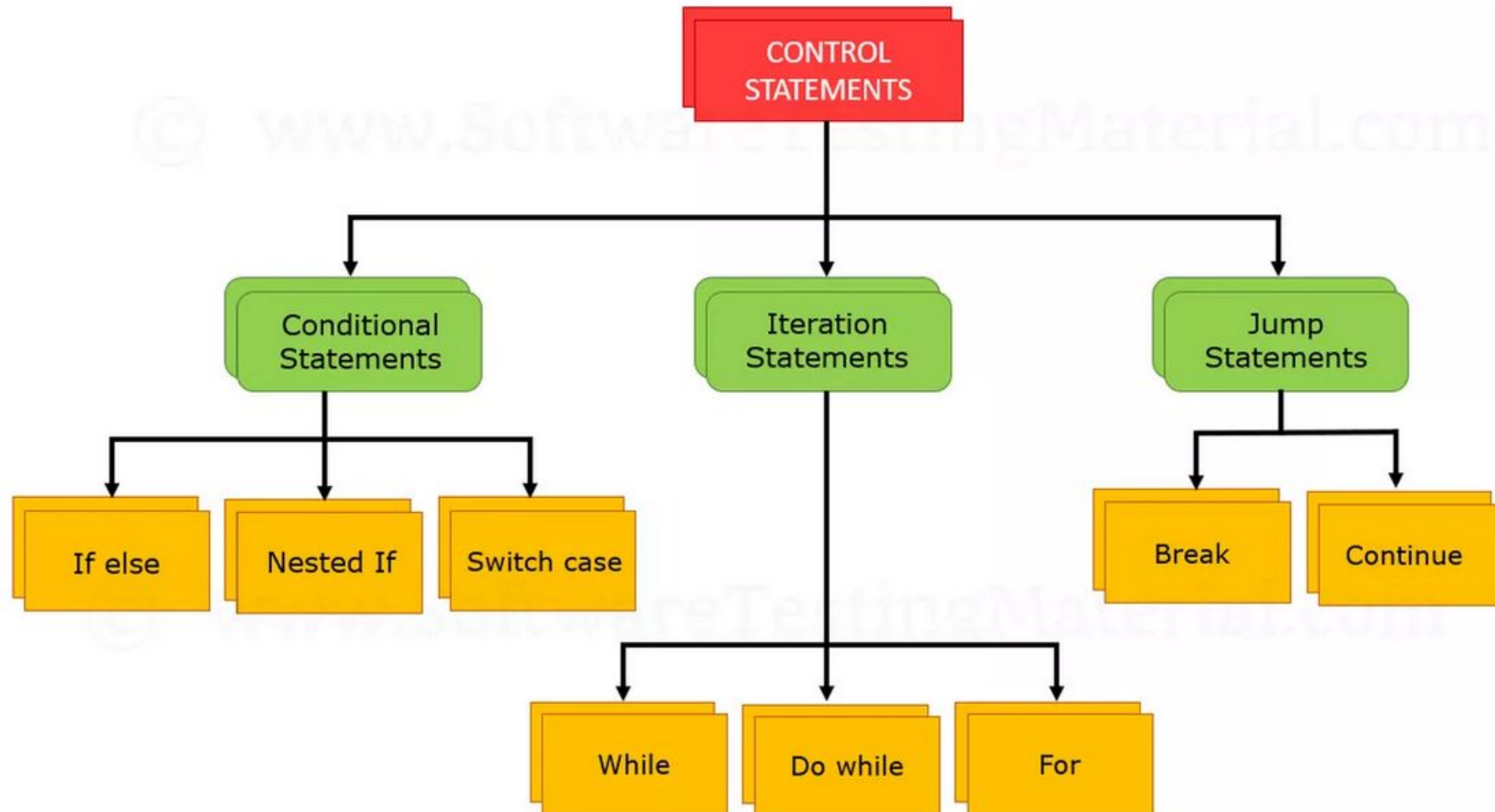
```
#include <iostream>
using namespace std;

int main() {

    int a = 12;
    int b = 10;

    cout << "a > b " << (a>b) << endl;
    cout << "a >= b " << (a>=b) << endl;
    cout << "a < b " << (a<b) << endl;
    cout << "a <= b " << (a<=b) << endl;
    cout << "a == b " << (a==b) << endl;
    return 0;
}
```

Control de flujos



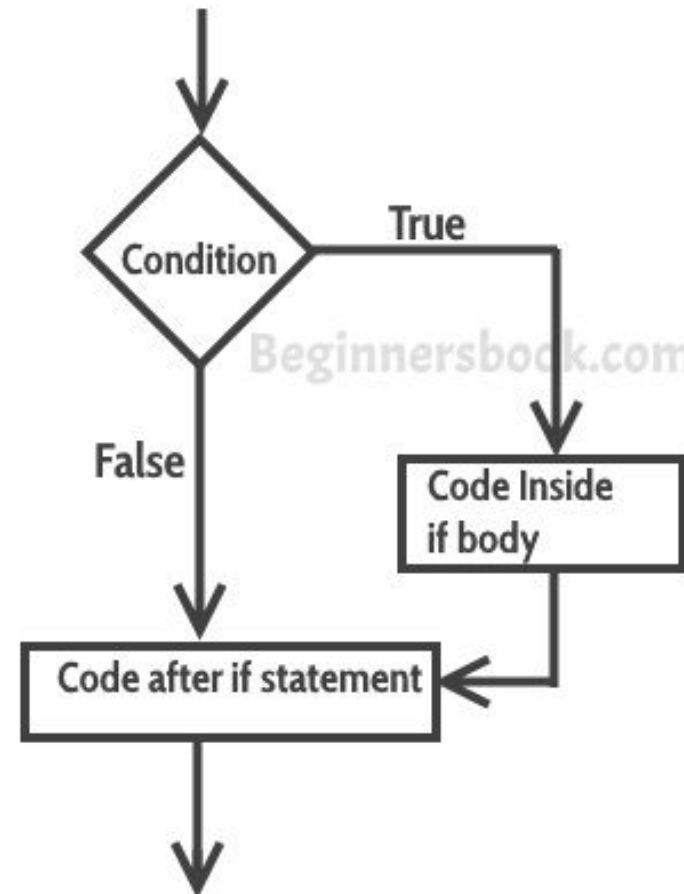
Declaraciones condicionales

- Declaración if
- Declaración if (anidado)
- Declaración if-else
- declaración if-esle-if

Declaraciones condicionales

- Declaración if

```
if(condition){  
    Statement(s);  
}
```



Declar

les

- Declaración if

```
#include <iostream>
using namespace std;
int main(){
    int num=70;
    if( num < 100 ){
        /* This cout statement will only execute,
        * if the above condition is true
        */
        cout<<"number is less than 100";
    }

    if(num > 100){
        /* This cout statement will only execute,
        * if the above condition is true
        */
        cout<<"number is greater than 100";
    }
    return 0;
}
```

Declaraciones condicionales

- Declaración if Anidado

```
if(condition_1) {  
    Statement1(s);  
  
    if(condition_2) {  
        Statement2(s);  
    }  
}
```

```
#include <iostream>  
using namespace std;  
int main(){  
    int num=90;  
    /* Nested if statement. An if statement  
     * inside another if body  
     */  
    if( num < 100 ){  
        cout<<"number is less than 100"<<endl;  
        if(num > 50){  
            cout<<"number is greater than 50";  
        }  
    }  
    return 0;  
}
```

Declaraciones condicionales

- Declaración if Anidado

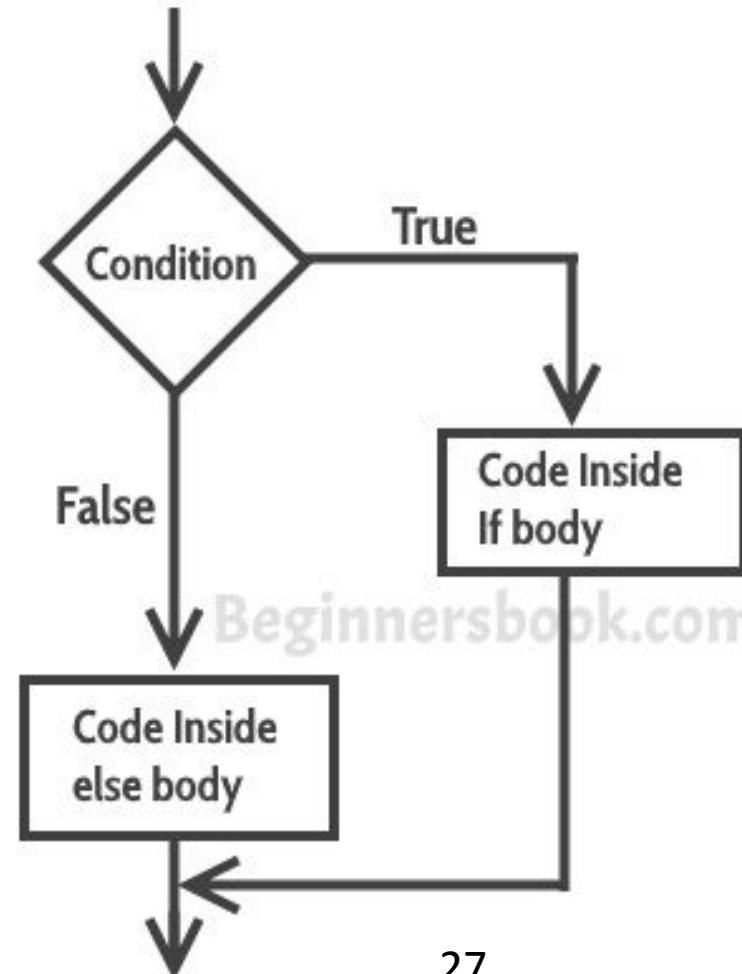
```
if(condition_1) {  
    Statement1(s);  
  
    if(condition_2) {  
        Statement2(s);  
    }  
}
```

```
#include <iostream>  
using namespace std;  
int main(){  
    int num=90;  
    /* Nested if statement. An if statement  
     * inside another if body  
     */  
    if( num < 100 ){  
        cout<<"number is less than 100"<<endl;  
        if(num > 50){  
            cout<<"number is greater than 50";  
        }  
    }  
    return 0;  
}
```

Declaraciones condicionales

- Declaración if-else

```
if(condition) {  
    Statement(s);  
}  
else {  
    Statement(s);  
}
```



Declaraciones condicionales

- Declaración if-else

```
#include <iostream>
using namespace std;
int main(){
    int num=66;
    if( num < 50 ){
        //This would run if above condition is true
        cout<<"num is less than 50";
    }
    else {
        //This would run if above condition is false
        cout<<"num is greater than or equal 50";
    }
    return 0;
}
```


Declaraciones condicionales

- Declaración if-else-if

```
#include <iostream>
using namespace std;
int main(){
    int num;
    cout<<"Enter an integer number between 1 & 99999: ";
    cin>>num;
    if(num <100 && num>=1) {
        cout<<"Its a two digit number";
    }
    else if(num <1000 && num>=100) {
        cout<<"Its a three digit number";
    }
    else if(num <10000 && num>=1000) {
        cout<<"Its a four digit number";
    }
    else if(num <100000 && num>=10000) {
        cout<<"Its a five digit number";
    }
    else {
        cout<<"number is not between 1 & 99999";
    }
    return 0;
}
```


Declaraciones condicionales

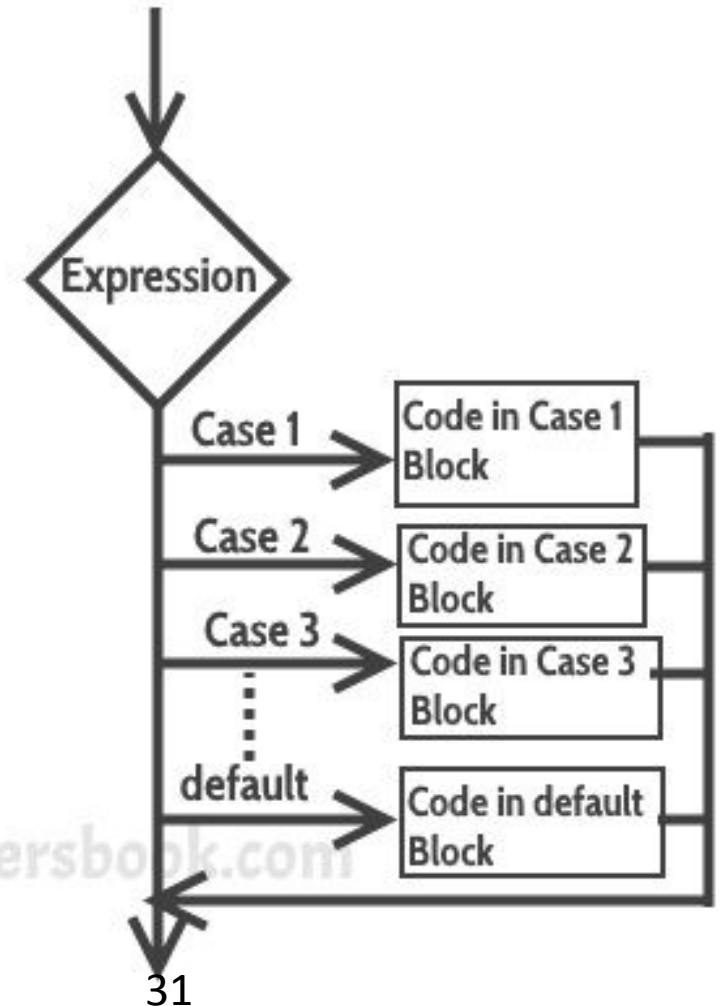
Ejercicios:

1. Determinar si un número es positivo o negativo.
2. Determinar si un número es par o impar.

Para resolver el ejercicio tienen **20 minutos**

Switch Case

```
switch (variable or an integer expression)
{
    case constant:
        //C++ code
        ;
    case constant:
        //C++ code
        ;
    default:
        //C++ code
        ;
}
```



Switch Case

```
#include <iostream>
using namespace std;
int main(){
    int num=5;
    switch(num+2) {
        case 1:
            cout<<"Case1: Value is: "<<num<<endl;
        case 2:
            cout<<"Case2: Value is: "<<num<<endl;
        case 3:
            cout<<"Case3: Value is: "<<num<<endl;
        default:
            cout<<"Default: Value is: "<<num<<endl;
    }
    return 0;
}
```

```
#include <iostream>
using namespace std;
int main(){
    int i=2;
    switch(i) {
        case 1: cout<<"Case1 " <<endl;
        case 2: cout<<"Case2 " <<endl;
        case 3: cout<<"Case3 " <<endl;
        case 4: cout<<"Case4 " <<endl;
        default: cout<<"Default " <<endl;
    }
    return 0;
}
```

Switch Case

Ejercicios:

1. Desarrollar una calculadora usando Switch Case.

Bucles

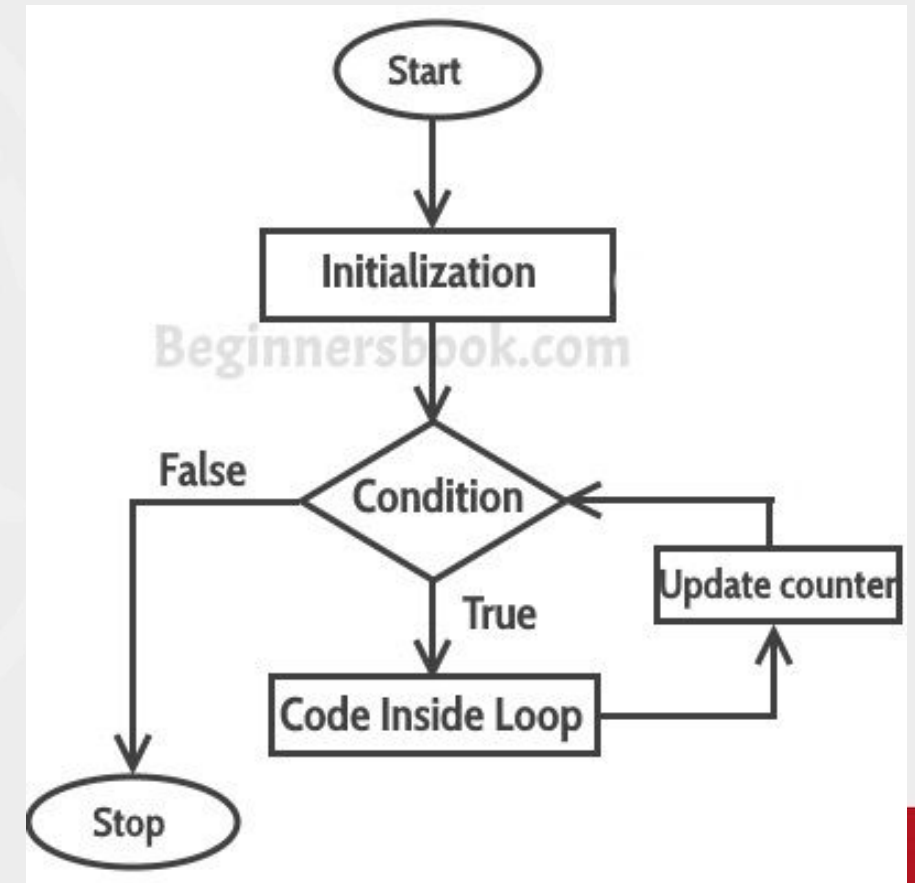
Declaraciones que repiten una instrucción N cantidad de veces, hay tres tipos:

1. for
2. while
3. do-while

Bucle For

Sintax:

```
for(initialization; condition ; increment/decrement)
{
    statement(s);
}
```



Bucle For

Sintax:

```
#include <iostream>
using namespace std;
int main(){
    for(int i=1; i<=6; i++){
        /* This statement would be executed
        * repeatedly until the condition
        * i<=6 returns false.
        */
        cout<<"Value of variable i is: "<<i<<endl;
    }
    return 0;
}
```


Bucle For

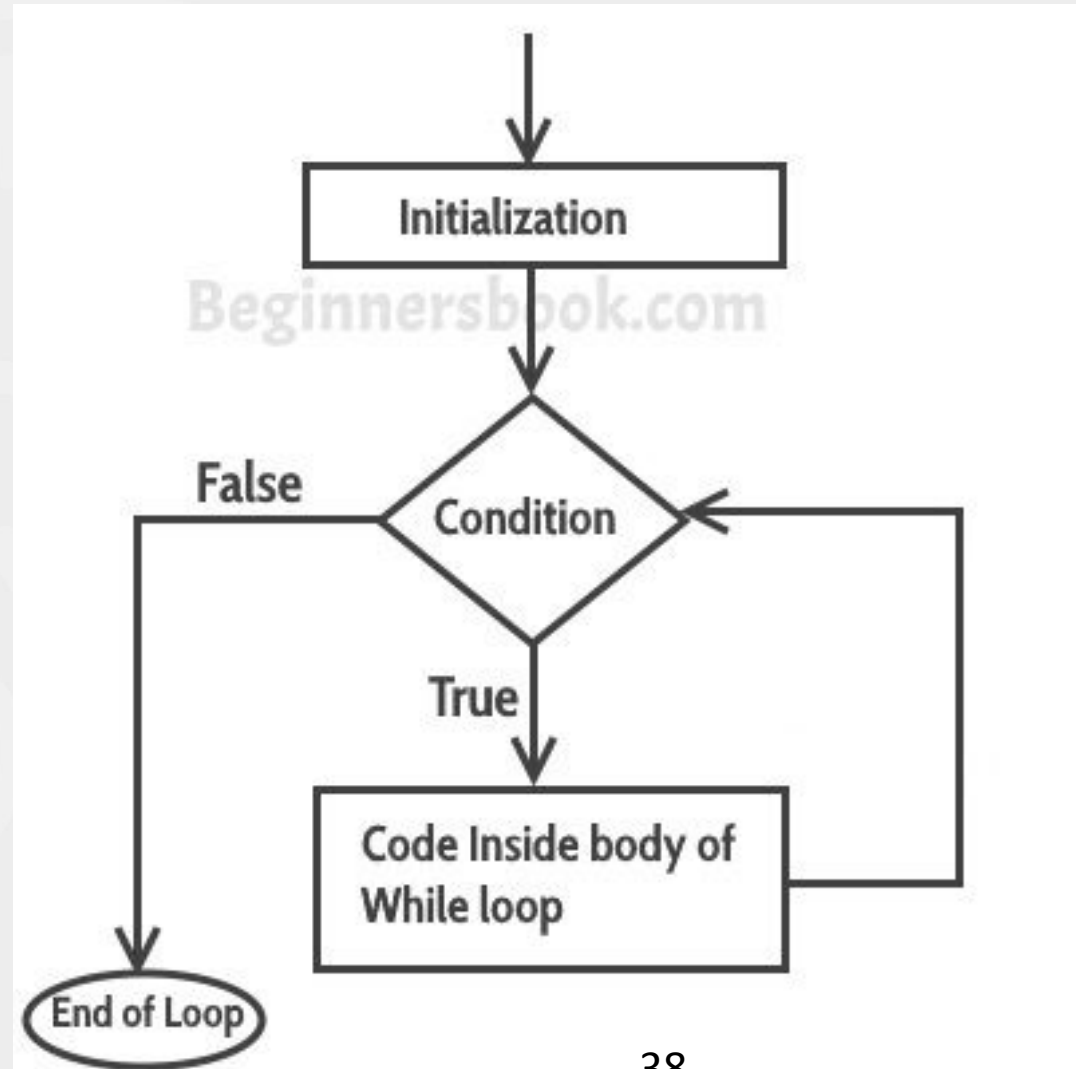
Sintax:

```
#include <iostream>
using namespace std;
int main(){
    int arr[]={21,9,56,99, 202};
    /* We have set the value of variable i
     * to 0 as the array index starts with 0
     * which means the first element of array
     * starts with zero index.
     */
    for(int i=0; i<5; i++){
        cout<<arr[i]<<endl;
    }
    return 0;
}
```

Bucle While

Sintax:

```
while(condition)
{
    statement(s);
}
```



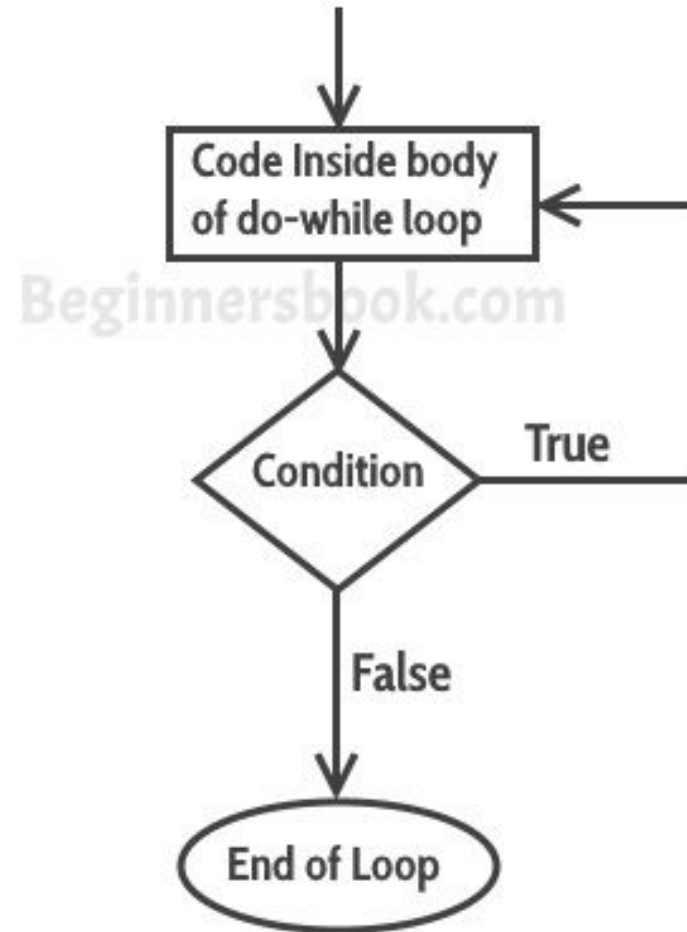
Bucle While

```
#include <iostream>
using namespace std;
int main(){
    int i=1;
    /* The loop would continue to print
     * the value of i until the given condition
     * i<=6 returns false.
     */
    while(i<=6){
        cout<<"Value of variable i is: "<<i<<endl; i++;
    }
}
```

Bucle Do-While

Sintax:

```
do  
{  
    statement(s);  
} while(condition);
```



Bucle Do-While

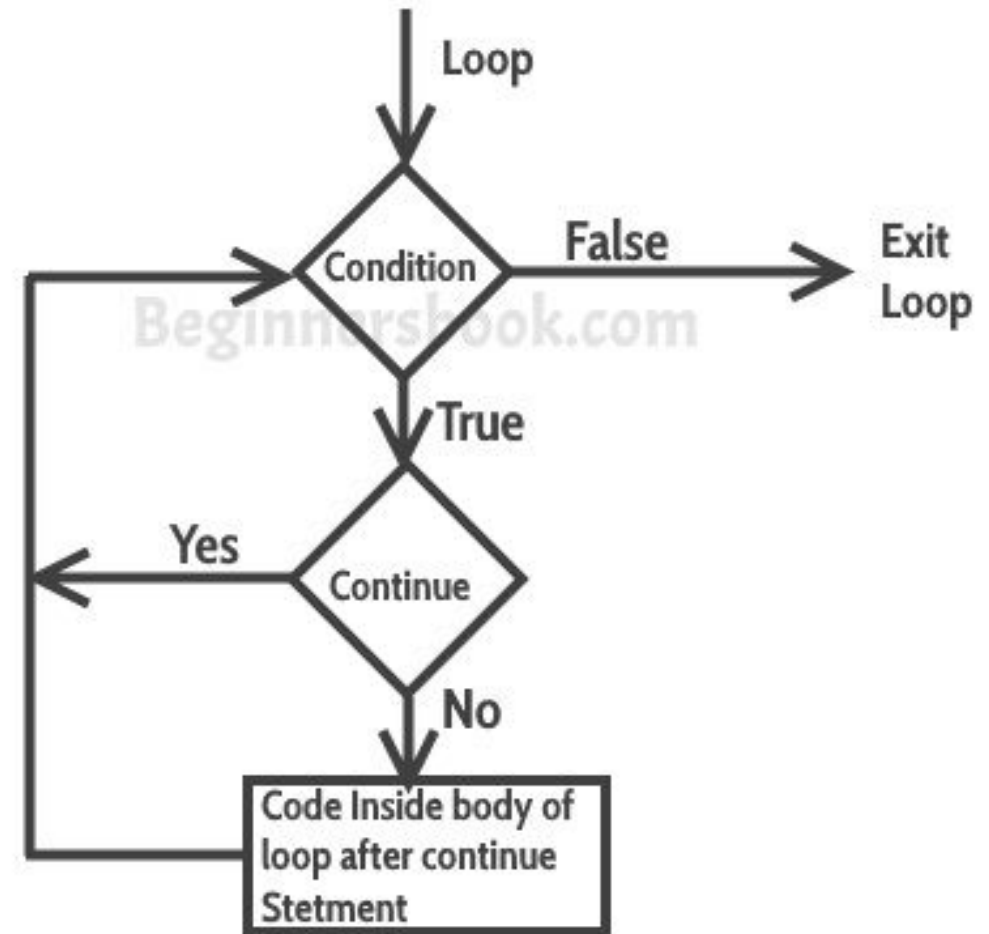
```
#include <iostream>
using namespace std;
int main(){
    int num=1;
    do{
        cout<<"Value of num: "<<num<<endl;
        num++;
    }while(num<=6);
    return 0;
}
```



Declaración continue

Sintax:

```
continue;
```



Declaración continue

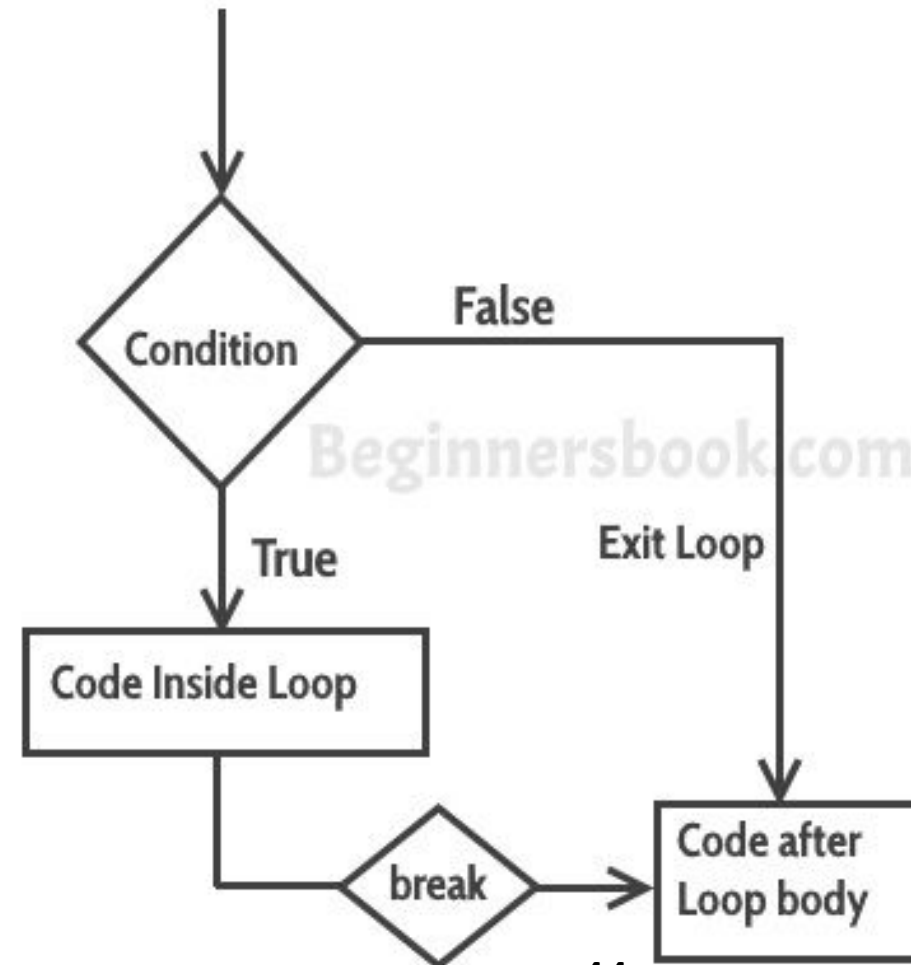
```
#include <iostream>
using namespace std;
int main(){
    int j=6;
    while (j >=0) {
        if (j==4) {
            j--;
            continue;
        }
        cout<<"Value of j: "<<j<<endl;
        j--;
    }
    return 0;
}
```



Declaración break

Sintax:

```
break;
```



Declaración break

```
#include <iostream>
using namespace std;
int main(){
    int num =10;
    while(num<=200) {
        cout<<"Value of num is: "<<num<<endl;
        if (num==12) {
            break;
        }
        num++;
    }
    cout<<"Hey, I'm out of the loop";
    return 0;
}
```



Declaración goto

Sintax:

```
goto label_name;
```

Program structure:

```
label1:  
...  
...  
goto label2;  
...  
..  
label2:  
...
```



**UNIVERSIDAD
DE LOS LLANOS®**

Declaración goto

```
#include <iostream>
using namespace std;
int main(){
    int num; cout<<"Enter a number: "; cin>>num;
    if (num % 2==0){
        goto print;
    }
    else {
        cout<<"Odd Number";
    }

    print:
    cout<<"Even Number";
    return 0;
}
```



Bucles

Ejercicios:

1. Encontrar el Factorial de un número, hacerlo tanto para For y While.

Referencias

Los ejercicios fueron tomados de

<https://beginnersbook.com/2017/08/c-plus-plus-tutorial-for-beginners/>

