

Lección 2

José David Ruiz Álvarez

josed.ruiz@udea.edu.co

Instituto de Física, Facultad de Ciencias Exactas y Naturales

Universidad de Antioquia

21 de agosto de 2018

1. Conceptos básicos de C++

1.1. Operadores

Operadores binarios: Requieren dos variables, por ejemplo para realizar la operación suma $a + b$.

- Suma: +
- Resta: -
- Multiplicación: *
- División: /
- Módulo: % (Solo se puede usar sobre enteros)

Operadores relacionales, ejemplo $i \leq 3$, estos operadores retornan un booleano *True* o *False* si la relación se cumple o no correspondientemente:

- Menor que: <
- Menor o igual que: <=
- Mayor que: >
- Mayor o igual que: >=
- Igual que: ==
- No igual que: !=

Operadores lógicos, se usan para combinar operaciones relacionales:

- OR: ||

- AND: `&&`
- Negación: `*`

Otros operadores:

- $A+ = 5 \leftrightarrow A = A + 5$
- $A- = 5 \leftrightarrow A = A - 5$
- $A* = 5 \leftrightarrow A = A * 5$
- $A/ = 5 \leftrightarrow A = A / 5$
- $A \% = 5 \leftrightarrow A = A \% 5$
- $A++ \leftrightarrow A = A + 1$
- $A-- \leftrightarrow A = A - 1$

1.2. Tipos de variables

Esta es una de las diferencias fundamentales con Python en donde no se requiere declarar el tipo de una variable para asignarle un valor. En C++ es necesario declarar el tipo de una variable para asignarle un valor, sin embargo ambas acciones se pueden hacer en una sola línea. Los tipos de variables básicos (y más usados, pero hay más!) en C++ son:

- `double`: Número real de 15 dígitos + signo
- `float`: Número real de 7 dígitos + signo
- `int`: Número entero de 10 dígitos + signo
- `char`: Un solo caracter.
- `bool`: Booleano “True” o “False”

El manejo de strings en C++ es poco intuitivo. Si necesitan hacer un programa para hacer algún tipo de operaciones sobre strings es aconsejable usar python en lugar de C++.

1.3. Bucles y condicionales

Ejemplo de condicional simple:

```
int a = 12;
if (a<=30){
cout << a*5 << endl;
}
```

Ejemplo de condicional y “else”:

```
int a = 12;
int b = 13;
if (a<=30 && b>30){
cout << "Both conditions satisfied" << endl;
}
else {
cout << "No condition satisfied" << endl;
}
```

Ejemplo de condicionales anidados:

```
float a = 12.53;
if (a<=2){
cout << "First condition satisfied" << endl;
}
else if (a<=6 && a>2){
cout << "Second condition satisfied" << endl;
}
else if (a<=13 && a>6){
cout << "Third condition satisfied" << endl;
}
```

Ejemplo de “while”:

```
double b = 1.001;
while (b<10000)
```

```
{  
b*=1.001;  
cout << b << endl;  
}
```

Ejercicio 1: Implemente el mismo ejemplo de while de C++ en python y compare el tiempo que toma hacer el cálculo en ambos lenguajes. Suba los resultados a github en forma de pull request en la carpeta “Ejercicio1_Leccion2”. Recuerde poner su nombre en el nombre de los archivos.

Ejemplo de “for”:

```
double b = 1.001;  
double c = 1.001;  
for (int i=0; i<20; i++)  
{  
    b*=(c+i);  
    cout << b << endl;  
}
```

Ejercicio 2: Haga un script de C++ que calcule los números de fibonacci y que imprima en pantalla únicamente los números pares de la serie. Guarde los resultados en un archivo de texto. Suba el script y los resultados a github en forma de pull request en la carpeta “Ejercicio2_Leccion2”. Recuerde poner su nombre en el nombre de los archivos.