

# Funciones y Clases de C++

José David Ruiz Álvarez

josed.ruiz@udea.edu.co

Instituto de Física, Facultad de Ciencias Exactas y Naturales

**Universidad de Antioquia**

10 de marzo de 2020

## 1. Funciones en C++

Toda función en C++ está definida por: Un tipo (“como” los tipos de variables), un nombre, parámetros y el cuerpo. El tipo puede ser “double”, “float”, etc. Pero además hay algunos tipos especiales para funciones:

- void: Una función que no retorna ningún valor
- inline: Una función, muy corta, que no es estrictamente llamada pero que su código es insertado donde se llama.

### Ejemplo 1:

```
int SumaSimple(int a, int b) //int es el tipo de la funcion, SumaSimple su nombre, y a y b sus parámetros
{
    //aquí empieza el cuerpo de la funcion
    int c = a+b;
    return c;
    //aquí termina el cuerpo de la funcion
}
```

### Ejemplo 2:

```
void PrintSumaSimple(int a, int b)
{
    int c = a+b;
    cout << c << endl;
}
```

### Ejemplo 3:

```
inline int SumaSimpleI(int a, int b)
{
    return a+b;
}
```

Los parámetros de una función en C++ pueden ser pasados a una función por su valor o por su referencia, en el primer caso la función se evalúa sobre los valores que le son entregados, en el segundo caso la función se evalúa sobre los valores que tienen las referencias que le son entregadas, además si la variable que es entregada por referencia cambia de algún modo dentro de la función el cambio es visible también afuera de la función.

#### **Ejemplo 4:**

```
void SumaAlPrimero(float &a, float b)
{
    a = a+b;
}
```

**Ejercicio 1:** Consultar la definición de protofunciones y la utilización de funciones por recursividad.

## **2. Clases**

### **2.1. Estructuras de datos**

Las estructuras de datos permiten conjugar o combinar en un solo objeto diferentes tipos de variables. Las estructuras de datos podemos pensarlas como el tipo más simple de clases en C++.

#### **Ejemplo 5:**

```
struct PosicionYCarga
{
    int Carga;
    float X;
    float Y;
    float Z;
} Particula1, Particula2;
```

## 2.2. Clases

La diferencia fundamental entre una estructura de datos y una clase es que esta última puede contener funciones. Además, dentro de una clase se pueden especificar los permisos de acceso a las componentes de la clase. El acceso a diferentes elementos de la clase puede ser público, privado o protegido. Cuando es público el acceso puede hacerse desde cualquier lugar donde el objeto de la clase sea visible. El acceso privado restringe el acceso a otros miembros dentro de la misma clase. Y el acceso protegido se restringe a miembros de la misma clase y clases derivadas.

### Ejemplo 6:

```
class Particle
{
public:
    int Carga;
    float X,Y,Z;
    float VX, VY, VZ;
    float M;
    void SetValues(float,float,float,float,float,float,float,int);
    float Pos_evol(float,float,float,float);
    //void time_slice_print();
} P1, P2;

void Particle::SetValues(float x,float y,float z,float vx,float vy,float vz,float m,int Car)
{
    X=x;
    Y=y;
    Z=z;
    VX=vx;
    VY=vy;
    VZ=vz;
    M=m;
    Carga=Car;
}
```

```
float Particle::Pos_evol(float ax, float ay, float az, float t)
{
X=X+(VX*t)+(0.5*ax*(t*t));
Y=Y+(VY*t)+(0.5*ay*(t*t));
Z=Z+(VZ*t)+(0.5*az*(t*t));
}
```