

Módulo Programación.

1º DAW.



PRÁCTICA 1: Funciones.

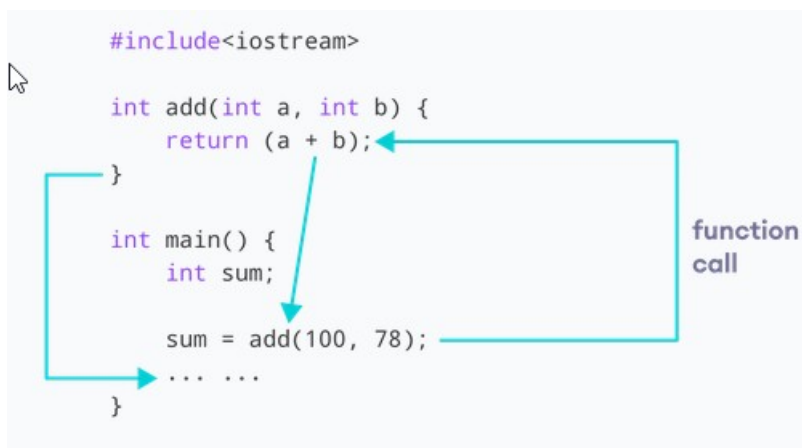
UNIDAD DE TRABAJO 4.

Profesor: Pedro Antonio Santiago Santiago.

1. Introducción.

Uno de los pilares de la programación estructurada y de la programación funcional es el concepto de función, heredado de su definición en matemáticas. Las funciones siguen utilizando ampliamente en la actualidad, tanto en los lenguajes estructurados, en especial C como en otros más modernos como PHP o Javascript.

En esta práctica se trabaja la definición, creación y uso de funciones, junto con el paso de parámetros por valor y por referencia.



```
#include<iostream>

int add(int a, int b) {
    return (a + b);
}

int main() {
    int sum;

    sum = add(100, 78);
    ...
}
```

2. Materiales.

En el caso de C/C++ los compiladores más usados son gcc y g++, disponibles en Linux, en Windows existen otros el propio de Microsoft o Mingw que emula el terminal de Linux.

Dado que a lo largo del curso solo se va a utilizar C/C++ en esta práctica se opta por usar un compilador y entorno online, seleccionando la web.

https://www.onlinegdb.com/online_c++_compiler

Existen muchas otras páginas web que permiten el desarrollo con C/C++, aunque en esta se programa siguiendo C++.

3. Desarrollo de la práctica.

Es imprescindible documentar el código.

La práctica pesa un 20% de la parte de las prácticas, que a su vez es un 40% de la nota de la unidad.

3.1. Ejercicios.

1. Crear una función llamada mayor que indique si un número entero es mayor que otro.
2. Realizar 2 llamadas a la función anterior con los valores 4,5 y 6,2.
3. Crear una función llamada mayor que devuelve el mayor de 3 caracteres en el mismo fichero que la función del ejercicio 1.
4. Usar la función del ejercicio 3 con los números 5,6,1 y 4,88,64.
5. Crear una función que descomponga un número en factores primos, es necesario además crear otra función que indique si un número es primo o no, por ejemplo al descomponer 45, se empieza por el 2 que es primo (llamara a esprimo(2)) pero 45 no es divisible por 2, pasando al 3, que es primo (se comprueba) y 45 es divisible por 3, quedando 15 y mostrando el 3, a continuación se pasa a 4 y se comprueba si es primo (no lo es), con lo que se pasa al 5....
6. Probar la función anterior con los números 15, 56 y 678.
7. Crear una función que devuelve el área de un círculo con valores reales y probarlo.
8. Implementar una función que imprima un vector de enteros, recordar que no se tiene el atributo length en C.
9. Desarrollar una función que ordene un vector, llamar a la función e imprimir a continuación el vector ordenado.
10. Sin escribir el código de la siguiente función indicar que valen las variables pasadas como parámetros al salir de la misma.

```
#include <iostream>

using namespace std;
void f(int ,int );
```

```
int main()
{
    int a=4;
    int b=7;
    f(a,b);
    cout<<a<<" "<<b<<"\n";
    return 0;
}
void f(int a, int b){
    a++;
    b--;
}
```

11. Realizar el mismo análisis que en el ejercicio anterior.

```
#include <iostream>

using namespace std;
void f(int &,int &);

int main()
{
    int a=4;
    int b=7;
    f(a,b);
    cout<<a<<" "<<b<<"\n";
    return 0;
}
void f(int &a, int &b){
    a++;
    b--;
}
```

12.. Analizar las diferencias entre los dos ejercicios anteriores.

13. Crear una función recursiva que calcule eleve un número a otro y probar. Definir en primer lugar caso base y caso recursivo en papel.

14. Definir una función que indique si un número es primo o no, de forma recursiva. Definir en primer lugar caso base y caso recursivo en papel.

4. Entrega.

La práctica se entrega en formato ZIP con 2 ficheros (código y documento en formato PDF, en el campus virtual ww.aules.edu.gva.es

. Se fijará la fecha en AULES.

El fichero 1 será el código generado comprimido a su vez también en zip, **importante comentar el código. Cada una de las partes en un proyecto Maven.**

El fichero 2 es el documento de entrega ha de tener los siguientes puntos.

1. Portada. (Título de la práctica y autor).
2. Introducción.
3. Desarrollo de la práctica. **Documento de texto con el código fuente comentado, capturas de ejecución y respuestas a las preguntas.**
4. Conclusiones.

5. Evaluación.

Unos días después de la entrega se realizará la corrección de forma presencial donde el profesor preguntará cuestiones sencillas sobre la práctica para comprobar la autoría de la misma.

RA2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos..

- CE2f. Se han utilizado parámetros en la llamada a métodos.
- CE2f. Se han utilizado parámetros en la llamada a métodos. .
- CE2i. Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.