

Universidad Nacional de Ingeniería
Centro de Tecnologías de Información y
Comunicaciones
CTIC—UNI

Programación en C++

Sesión 4a - Funciones

Funciones

- Definición:
 - Las funciones son los bloques de construcción básicos de C++. Dentro de ellas se da toda la actividad del programa.
- Criterios para crearlas:
 - Se usa la estrategia de “Dividir y Conquistar”, en otras palabras, es realizar la partición de la tarea en subtareas más fáciles de abordar.
 - Todo problema se puede resolver con funciones o sin ellas, pero su uso adecuado hace que el programa sea más eficiente, fácil de leer y probar.

Funciones: ventajas

- 1.- Facilita el diseño descendente.
- 2.- Los procesos dentro de ellas se pueden ejecutar varias veces.
- 3.- Facilita la división de tareas.
- 4.- Se pueden probar individualmente.
- 5.- Con funciones apropiadamente diseñadas, es posible ignorar como se realiza una tarea, sabiendo qué es lo que hacen.

Funciones: Especificación formal

```
tipo nombre_funcion(lista de parámetros){  
    cuerpo de la función  
}
```

- **tipo** : especifica el tipo de valor que devuelve la función. Si no se especifica tipo, el compilador asume que es entero (int).
- **lista de parámetros** : es la lista de nombres de variables separados por comas con sus tipos asociados que reciben los valores de los argumentos actuales de la llamada a la función.

La sentencia *return*

- Fuerza la salida inmediata de la función en que se encuentra.
- Se puede utilizar para devolver un valor.

Ejemplo

Construya un programa que calcule el cuadrado de los 10 primeros números enteros.

Solución

```
#include <iostream>

int cuadrado(int x) {
    int r;
    r=x*x;
    return r;
}

...
```

Solución

```
#include <iostream>
```

```
int cuadrado(int x){  
    int r;  
    r=x*x;  
    return r;  
}
```

```
int main(){  
    int i,resultado;  
    for (i=0;i<10;i++){  
        resultado = cuadrado (i);  
        cout<<"Cuadrado de "<<i<<"= "<<resultado;  
    }  
    return 0;  
}
```


Ejemplo

Construya una función que calcule la suma de los k primeros números enteros impares.

$$S = \sum_{i=1}^k 2i + 1$$

Solución ?

```
#include <iostream>

int sumaImpares (-----) {
    -----
    return r;
}
```

Ejercicio

1. Escribir una función Salario que calcule los salarios de un trabajador para un número dado de horas trabajadas y un salario hora. Las horas que superen las 40 horas secuenciales se pagarán como extras con un salario hora 1,5 veces el salario ordinario.

2. Elabore una función que pida un entero de cualquier cantidad de cifras y retorne la suma de dígitos.

Ejercicio

Dado el valor de un ángulo, escribir una función que muestre el valor de todas las funciones trigonométricas correspondientes al mismo.

Ejercicio

Escriba una función múltiplo que determine para un par de enteros, si el segundo es múltiplo del primero. La función debe tomar dos argumentos enteros y devolver 1 (verdadero) si el segundo es un múltiplo del primero, y de lo contrario 0 (falso). Utilice esta función en un programa que introduzca una serie de pares de enteros.

Funciones

- Las funciones terminan su ejecución ya sea a través de la ejecución de la última instrucción o utilizando la sentencia `return`.
- Una función puede retornar valor sólo cuando el tipo de retorno no es `void`.
- Las funciones que son de tipos distintos de `void` pueden ser utilizadas como operandos en otras expresiones del lenguaje.

Ejercicio

Un número primo es una cantidad entera positiva únicamente divisible por uno o por sí mismo.

Escribir una función entera **primo** que retorne 1 si el número a evaluar es un primo y 0 en caso contrario.

Desde la función principal llame a la función **primo** desde una estructura selectiva.

Funciones: modos de uso

- 1.- Funciones diseñadas para realizar operaciones a partir de sus argumentos y devolver un valor basado en sus cálculos.
- 2.- Funciones que no reciben argumentos, realizan un proceso y devuelven un valor.
- 3.- Funciones que no tienen argumentos ni valor de retorno explícito, operan sobre el entorno de variables globales o atributos del sistema operativo.

Función *main()*

- Mediante la sentencia `return`, el programa puede devolver un código de terminación al proceso de llamada (**Ej.:** entero a Shell de comandos en Unix).
- El valor devuelto puede ser 0 que indica terminación normal o un valor que identifica un error detectado por el programa.
- El valor devuelto puede ser usado para ejecución condicional de comandos en Shell en Unix.