UNIVERSIDAD CATÓLICA
de Colombia
Vigilada Mineducación

Procedural Programming
Basic Concepts

Diego Alberto Rincón Yáñez
darincon@ucatolica.edu.co

Definición de Función

- Conjunto de Instrucciones que realizan una tarea Específica
- En general toman valores de entrada, llamados parámetros
- En general proporcionan un valor de salida, llamado retorno

Ejemplo de función que…

- No reciba nada como parámetro
- Retorne un número entero
- El numero entero que retorna es diferente cada vez.

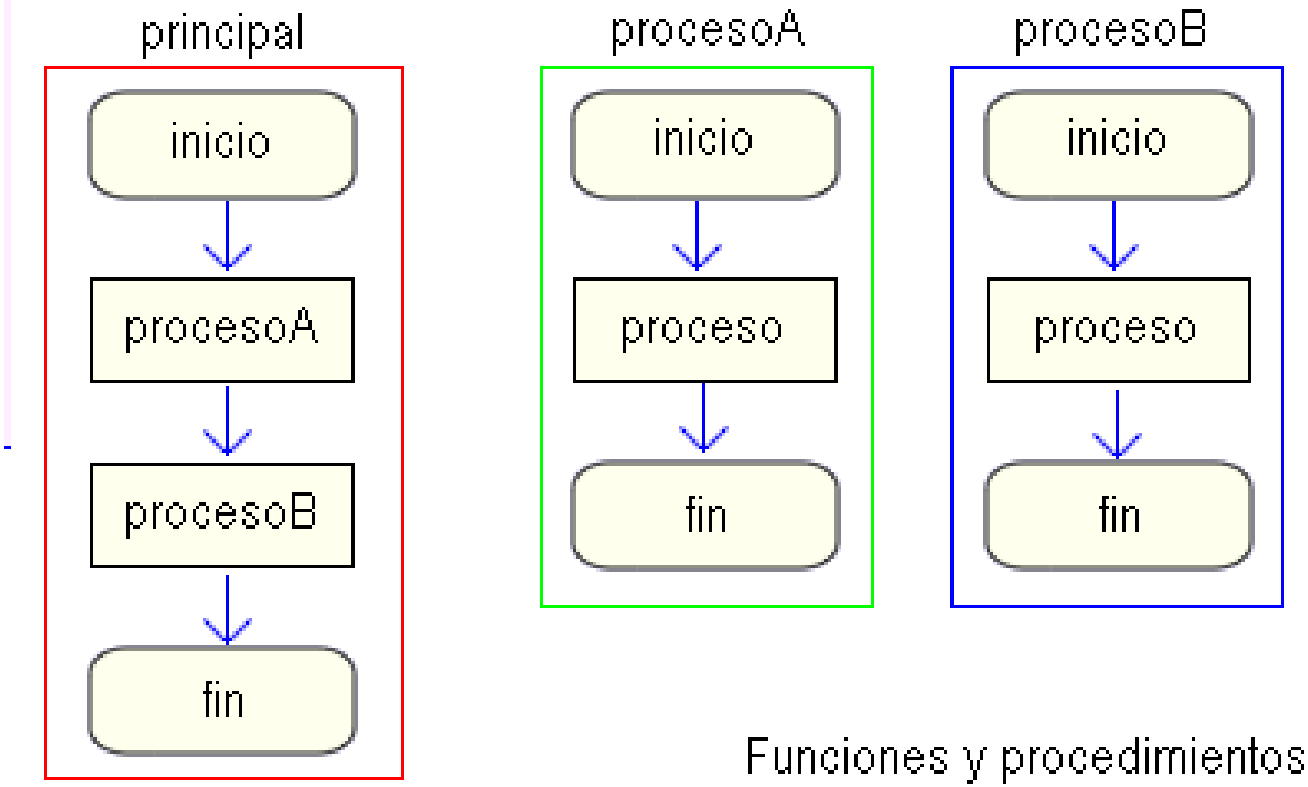
Ejemplo de función

- LA FUNCIÓN RAND!
  - *rand*()…

Para crear nuestras propias funciones

- Es posible realizar funciones definidas por el programador
- Para definir funciones es necesario seguir unos pasos

Subrutinas en C++



Funciones y procedimientos

Estructura del programa

- [directivas del pre-procesador: #include y #define]
- [prototipos de funciones]
- función main
- [definiciones de funciones]

Paso 1: Definición de Prototipos

- [directivas del pre-procesador: #include y #define]
- [prototipos de funciones]
- función main
- [definiciones de funciones]

Paso 1: Definición de Prototipos

- [prototipos de funciones]
- Los Prototipos de funciones (o firma de función) contiene, principalmente 4 partes
- [Valor de retorno] *[nombre de la función]* ([Lista de parámetros]);
- Ejemplo:
- El prototipo de la función *rand* es
- int *rand* (void);

Paso 1: Definición de Prototipos

- [prototipos de funciones]
- La palabra void se utiliza para representar “vacío” o que no hay parámetros/tipo de retorno

Paso 1: Definición de Prototipos

- **Los Datos** que voy a evaluar son los parámetros, éstos se separan por comas “,” en caso de existir más de uno
- **El tipo de dato** de lo que voy a responder es el retorno, solo puede haber UNO en todas las funciones

Ejercicio

- Supongamos una función que halla el mayor de dos números enteros:
- ¿Cuál(es) es/son el/los dato(s)?
- ¿Cuál es el tipo de retorno?
- ¿Cómo es el nombre?
- ¿Cómo es el prototipo de dicha función?

Respuesta

- Supongamos una función que halla el mayor de dos números enteros:
- ¿Cuál(es) es/son el/los dato(s)?
  - Numero Entero a y numero Entero b
- ¿Cuál es el tipo de retorno?
  - Un Entero (Ya que voy a responder bien sea con el Numero Entero a o con el Numero Entero b)

Respuesta

- Supongamos una función que halla el mayor de dos números enteros:
- ¿Cuál(es) es/son el/los dato(s)?
  - Numero **Entero** a y numero **Entero** b
- ¿Cuál es el tipo de retorno?
  - Un **Entero** (Ya que voy a responder bien sea con el Numero **Entero** a o con el Numero **Entero** b)

Reglas para nombramiento

- Decidir el nombre:
  - Reglas:
    - El nombre no debe contener espacios
    - Debe ser lo más descriptivo posible
    - Pueden contener el carácter “_”
    - Se recomienda la notación que indica colocar la primera letra de una palabra en Mayuscula
  - Ejemplos validos:
    - *mayor*
    - *Mayor_de_dos_numeros*
    - *mayorDeDosNumeros*

Respuesta

- Uniendo todo
- Tipo de retorno
  - Un **Entero** (Ya que voy a responder bien sea con el Numero **Entero** a o con el Numero **Entero** b)
- Nombre
  - *mayor*
- Datos
  - Numero **Entero** a y numero **Entero** b

Respuesta

- Uniendo todo

- **int** *mayor* (**int** a, **int** b);

Paso 2: Definición de Funciones

- [directivas del pre-procesador: #include y #define]
- [prototipos de funciones]
- función main
- [definiciones de funciones]

Paso 2: Definición de Funciones

- Debe iniciar con el prototipo, cambiando el ; por {
- Dentro de la definición de cada función está el "sub-programa" que va a realizar las operaciones.
- **SIEMPRE** se finaliza con la palabra **return** dato; donde dato es el nombre de la variable que se va a retornar.
- En caso de tener tipo de retorno **void** no es necesario colocar la palabra **return** sin embargo, es mejor colocarla al final de esta manera: **return;**

Ejercicio

- ¿Cómo sería el “sub-programa” de la función *mayor* ? (Suponiendo que ya tenemos los números a y b, es decir, no hay que declararlos ni pedirlos)

Respuesta

- …Suponiendo que tenemos los números a y b

```
if (a<b){
    return b;
}else{
    return a;
}
```

REGLA: Cuando Una Función Llega A Un return, La Función Termina

Ejercicio

- ¿Cómo sería la definición de la función *mayor*?

Respuesta

- ¿Cómo sería la definición de la función *mayor*?

```
int mayor (int a, int b){
    if (a<b){
        return b;
    }else{
        return a;
    }
}
```

Paso 3: Llamar la función

- [directivas del pre-procesador: #include y #define]
- [prototipos de funciones]
- función main
- [definiciones de funciones]

Paso 3: Llamar la función

- Las funciones se llaman dentro de la función *main*
- Ejemplo:

```
//….Contando con a y b….
int numMayor;
numMayor = mayor(a,b);
```

Ejemplo

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

int mayor (int a, int b); //Declaración de la función o Prototipo

int main (){
    int num1,num2,numMayor;
    cout<<"Digite un numero: ";
    cin>>num1;
    cout<<"Digite otro numero: ";
    cin>>num2;
    numMayor = mayor(num1,num2); //Llamado a la función
    cout<<"El numero mayor es: "<<numMayor<<endl;
   system("pause");
   return 0;
}

int mayor (int a, int b){ //Definición de la función
     if (a<b) {
           return b;
     }else {
           return a;
     }
}
```

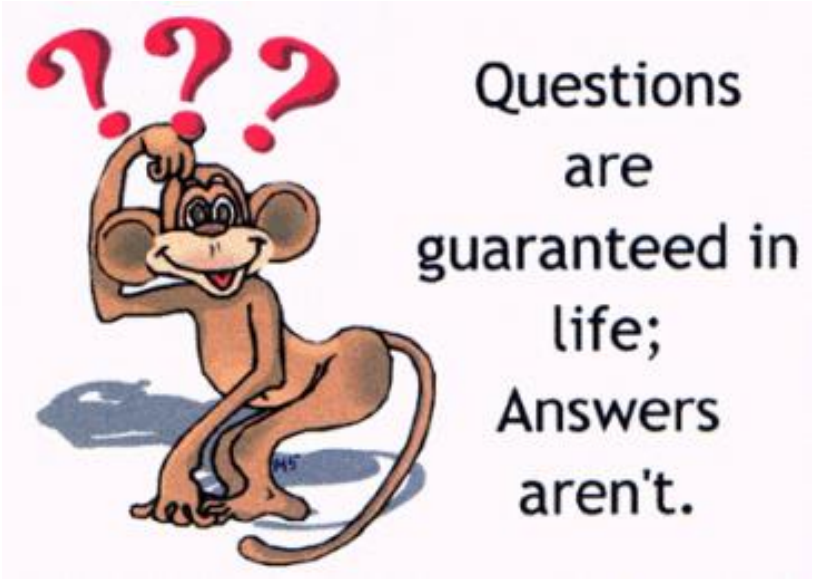
Ejemplo

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

int mayor (int a, int b); //Declaración de la función o Prototipo

int main (){
     int num1,num2,numMayor;
     cout<<"Digite un numero: ";
     cin>>num1;
     cout<<"Digite otro numero: ";
     cin>>num2;
     numMayor = mayor(num1,num2); //Llamado a la función
     cout<<"El numero mayor es: "<<numMayor<<endl;
    system("pause");
    return 0;
}

int mayor (int a, int b){ //Definición de la función
      if (a<b) {
            return b;
      }else {
            return a;
      }
}
```



¿Questions?

Diego Alberto Rincón Yáñez MCSc.
Twitter: @d1egoprog.