



UNIVERSIDAD CATÓLICA
de Colombia



UNIVERSIDAD CATÓLICA
de Colombia

Vigilada Mineducación

Procedural Programming

Basic Concepts

Diego Alberto Rincón Yáñez
darincon@ucatolica.edu.co



Funciones Especiales

- Para finalizar un programa y pausar
 - `system("pause");`
- Para hacer una nueva línea (equivalente a "`\n`")
 - `endl`
 - Uso: `cout<<...<<endl;`
- Para aumentar un numero
 - `numero++;`
 - `++numero;`



Diferencias

numero++

- Primero evalúa y después aumenta
- Ej:
- `a=0;`
- `while(a++!=1){...}`
 - ENTRA!!

++numero

- Primero aumenta y después evalúa
- Ej:
- `a=0;`
- `while(++a!=1){...}`
 - NO ENTRA!!



Variables Globales

- NO deben usarse frecuentemente
- Se ubican antes del main, por fuera de cualquier corchete
- Aunque las pueden ver todas las funciones, también las pueden modificar, lo cual es un riesgo muy grande



Variables Constantes

- Una variable constante es una variable definida que no puede cambiar en el tiempo es decir, el valor inicial que obtiene es el valor que se mantendrá.



Variables Constantes

- Sirve para declaración de vectores y matrices
- Se declara con la palabra **const** y por definición todas las letras del nombre son mayúsculas

const <tipoDato> <NOMBRE> = <valor>;

- Ejemplo:
 - `const int MAXF = 10;`



Manejo de variables

- Toda variable “vive” desde el corchete de apertura { donde se crea y el corchete final } una vez alcanzado ese corchete, nadie podrá “verla”



Manejo de variables

- Toda variable “vive” desde el corchete de apertura { donde se crea y el corchete final } una vez alcanzado ese corchete, nadie podrá “verla”



Manejo de variables

```
int main () {  
    int variableMain=0;  
    if (<condicion>){  
        int variableIF = 1;  
        variableMain++;  
    }  
    while (<condicion>){  
        bool variableWhile = true;  
        if (<condicion>){  
            int variableWhileIf = 0;  
            variableMain = 0;  
            variableWhile++  
        }  
    }  
    variableMain=10;  
}
```



Manejo de variables

```
int main () {  
    int variableMain=0;  
    if (<condicion>){  
        int variableIF = 1;  
        variableMain++;  
    }  
    while (<condicion>){  
        bool variableWhile = true;  
        if (<condicion>){  
            int variableWhileIf = 0;  
            variableMain = 0;  
            variableWhile++  
        }  
    }  
    variableMain=10;  
}
```



Manejo de variables

```
int main () {  
    int variableMain=0;  
    if (<condicion>) {  
        int variableIF = 1;  
        variableMain++;  
    }  
    while (<condicion>){  
        bool variableWhile = true;  
        if (<condicion>){  
            int variableWhileIf = 0;  
            variableMain = 0;  
            variableWhile++  
        }  
    }  
    variableMain=10;  
}
```

Fin variable
"variableIF"

Fin variable
"variableWhileIf"

Fin variable
"variableWhile"

Fin variable
"variableMain"



En Una Función...

```
int mayor ();  
int main () {  
    int numeroA=0, numeroB =0;  
    cout<<"Digite 2 numeros: "<<endl;  
    cout<<"Num1: ";  
    cin>>numeroA;  
    cout<<"Num2: ";  
    cin>>numeroB;  
}  
  
int mayor () {  
    if(numeroA > numeroB) {  
        return numeroA;  
    }else{  
        return numeroB;  
    }  
}
```



En Una Función...

```
int mayor ();
int main () {
    int numeroA=0, numeroB =0;
    cout<<"Digite 2 numeros: "<<endl;
    cout<<"Num1: ";
    cin>>numeroA;
    cout<<"Num2: ";
    cin>>numeroB;
}

int mayor() {
    if(numeroA > numeroB) {
        return numeroA;
    }else{
        return numeroB;
    }
}
```

Fin variables
"numeroA" y
"numeroB"

Las variables
"numeroA" y
"numeroB" NO
EXISTEN!



Solución (ERRONEA)

- “Declararlas dentro de la función!!!!”

```
int mayor() {  
    int numeroA, numeroB;  
    if(numeroA > numeroB) {  
        return numeroA;  
    } else {  
        return numeroB;  
    }  
}
```

Sin embargo se están volviendo a crear, no importa si se llaman o no igual, NO almacenan el valor que tenía en el main, ya que son 2 variables diferentes, sín importar que se llamen igual



Solución (CORRECTA)

- Mandarlos como parametro!

```
int mayor ();
int main () {
    int numeroA=0, numeroB =0;
    cout<<"Digite 2 numeros: "<<endl;
    cout<<"Num1: ";
    cin>>numeroA;
    cout<<"Num2: ";
    cin>>numeroB;
}

int mayor() {
    if(numeroA > numeroB) {
        return numeroA;
    }else{
        return numeroB;
    }
}
```



Solución (CORRECTA)

- Mandarlos como parametro!

```
int mayor (int numeroA, int numeroB) ;
int main () {
    int numeroA=0, numeroB =0, resp;
    cout<<"Digite 2 numeros: "<<endl;
    cout<<"Num1: ";
    cin>>numeroA;
    cout<<"Num2: ";
    cin>>numeroB;
    resp = mayor (numeroA, numeroB) ;
}

int mayor (int numeroA, int numeroB) {
    if (numeroA > numeroB) {
        return numeroA;
    } else {
        return numeroB;
    }
}
```



¿y Si cambio el nombre?

- Mandarlos como parametro!

```
int mayor (int num1,int num2);
int main () {
    int numeroA=0, numeroB =0, resp;
    cout<<"Digite 2 numeros: "<<endl;
    cout<<"Num1: ";
    cin>>numeroA;
    cout<<"Num2: ";
    cin>>numeroB;
    resp = mayor (numeroA, numeroB);
}


int mayor(int num1, int num2){
    if(num1> num2){
        return num1;
    }else{
        return num2;
    }
}
```



Solución (CORRECTA)

- Mandarlos como parametro!

```
int mayor (int num1, int num2);  
int main () {  
    int numeroA=0, numeroB =0, resp;  
    cout<<"Digite 2 numeros: "<<endl;  
    cout<<"Num1: ";  
    cin>>numeroA;  
    cout<<"Num2: ";  
    cin>>numeroB;  
    resp = mayor (numeroA, numeroB);  
}  
  
int mayor (int num1, int num2) {  
    if (num1 > num2) {  
        return num1;  
    } else {  
        return num2;  
    }  
}
```



The diagram consists of two blue arrows. One arrow originates from the 'numeroA' parameter in the function call 'mayor (numeroA, numeroB)' within the 'main' function and points to the 'num1' parameter in the function definition 'int mayor (int num1, int num2)'. The second arrow originates from the 'numeroB' parameter in the same function call and points to the 'num2' parameter in the function definition. This illustrates how the arguments are passed to the function.

SIEMPRE se mantiene el orden



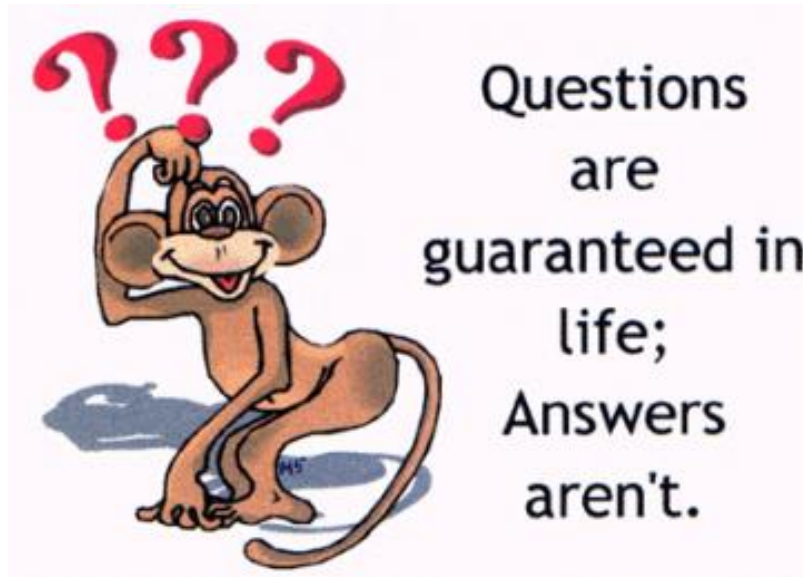
Cabe Anotar:

- En las variables enviadas el valor no cambia

```
int suma (int num1,int num2);
int main () {
    int numeroA=0, numeroB =0, resp;
    cout<<"Digite 2 numeros: "<<endl;
    cout<<"Num1: ";
    cin>>numeroA;
    cout<<"Num2: ";
    cin>>numeroB;
    resp = suma (numeroA, numeroB);
    cout<<"La suma entre "<<numeroA<<" y "<<numeroB<<"es: "<<resp;
}

int suma (int num1, int num2){
    num1 = num1+num2;
    cout<<"En la funcion num1 es: "<<num1;
    return num1;
}
```





¿Questions?

Diego Alberto Rincón Yáñez MCSc.
Twitter: @d1egoprog.

