

Funciones con parámetros



Por valor



Ejemplo 1 - Por valor

```

1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  void llamarFuncion(int valor);
7
8  int main()
9  {
10     int numero=10;
11
12     cout<<"Valor antes de la funcion: "<<numero<<endl;
13     llamarFuncion(numero);
14     cout<<"Valor despues de la funcion: "<<numero<<endl;
15
16     return 0;
17 }
18
19 void llamarFuncion (int valor)
20 {
21     cout<<"Valor del parametro que llega: "<<valor<<endl;
22     valor --;
23     cout<<"Valor despues de la operacion: "<<valor<<endl;
24 }

```

Para comprender mejor una función con parámetros por valor, se ilustra a continuación con un ejemplo:

- Línea 6: se escribe el prototipo de la función.
- Línea 10: se declara la variable local *numero* inicializada en 10.
- Línea 12: se imprime el valor de la variable local antes de que la función sea llamada.
- Línea 13: se llama la función con el parámetro *numero* que es igual a 10.
- Línea 14: se imprime el valor de la variable *numero* después de llamar la función.
- Línea 19: se inicia la definición de la función.
- Línea 21: se imprime el valor del parámetro que llega a la función.
- Línea 22: se efectúa una operación de decremento al parámetro que llegó a la función.
- Línea 23: se imprime el valor actual del parámetro.

Ejemplo 1 - Por valor

```

1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  void llamarFuncion(int valor);
7
8  int main()
9  {
10     int numero=10;
11
12     cout<<"Valor antes de la funcion: "<<numero<<endl;
13     llamarFuncion(numero);
14     cout<<"Valor despues de la funcion: "<<numero<<endl;
15
16     return 0;
17 }
18
19 void llamarFuncion (int valor)
20 {
21     cout<<"Valor del parametro que llega: "<<valor<<endl;
22     valor --;
23     cout<<"Valor despues de la operacion: "<<valor<<endl;
24 }

```

El orden de ejecución del programa anterior es el siguiente:

1. Se imprime el valor original de la variable *numero* (10).
2. Se hace el llamado a la función.
3. Se imprime el valor que llega como parámetro a la función (10).
4. Se hace la operación de decremento con el parámetro (10 - 1).
5. Se imprime el valor del parámetro después de la operación (9).
6. La función termina y entonces el programa regresa al punto en que esta fue llamada.
7. Se imprime el valor de la variable *numero* después de la función (10).

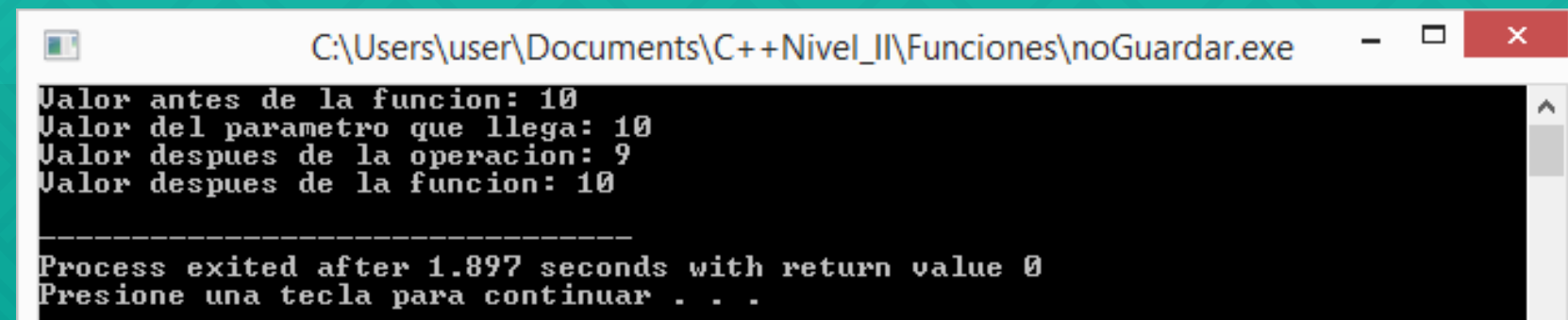
Ejemplo 1 - Por valor

```

1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  void llamarFuncion(int valor);
7
8  int main()
9  {
10     int numero=10;
11
12     cout<<"Valor antes de la funcion: "<<numero<<endl;
13     llamarFuncion(numero);
14     cout<<"Valor despues de la funcion: "<<numero<<endl;
15
16     return 0;
17 }
18
19 void llamarFuncion (int valor)
20 {
21     cout<<"Valor del parametro que llega: "<<valor<<endl;
22     valor --;
23     cout<<"Valor despues de la operacion: "<<valor<<endl;
24 }

```

Teniendo en cuenta este orden, el resultado devuelto al ejecutar el programa se observa a continuación:



```

C:\Users\user\Documents\C++Nivel_II\Funciones\noGuardar.exe
Valor antes de la funcion: 10
Valor del parametro que llega: 10
Valor despues de la operacion: 9
Valor despues de la funcion: 10

-----
Process exited after 1.897 seconds with return value 0
Presione una tecla para continuar . . .

```

Ejemplo 1 - Por valor

```

1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  void llamarFuncion(int valor);
7
8  int main()
9  {
10     int numero=10;
11
12     cout<<"Valor antes de la funcion: "<<numero<<endl;
13     llamarFuncion(numero);
14     cout<<"Valor despues de la funcion: "<<numero<<endl;
15
16     return 0;
17 }
18
19 void llamarFuncion (int valor)
20 {
21     cout<<"Valor del parametro que llega: "<<valor<<endl;
22     valor --;
23     cout<<"Valor despues de la operacion: "<<valor<<endl;
24 }

```

Como se puede ver el valor del parámetro se mantiene en la función *main* antes y después de la ejecución de la función *llamarFuncion*, dicho parámetro se utiliza para realizar la operación y sin embargo su valor inicial nunca se altera.

En conclusión, en el paso de parámetros por valor lo que se realiza entonces es enviar a la función una copia de la variable original para que sea utilizada con el fin de realizar acciones u operaciones dentro de la función a la que se le envía, por esta razón esa variable original se mantiene intacta y las funciones no tienen la propiedad de modificarla.