

# Lenguaje de Programación: C++, INPUT OUTPUT(Variables)

José Luis Alonzo Velázquez

Universidad de Guanajuato

Septiembre 2010

## MAIN

```
int main (); // no arguments
```

```
int main (int argc, char* argv[]); // arguments
```

argc: cantidad de parámetros contando el nombre del ejecutable que es el primer parámetro.

argv[]: Un arreglo que contiene todos los parámetros recibidos.

## Ejemplo

```
copy "C:\archivo.ext" "C:\User\archivo.ext"
```

el valor de `argc` = 3, ya que el nombre del programa es el primer parámetro y luego van los dos parámetros adicionales.

Como puedes apreciar `argv`

es un arreglo, de modo que, para este caso, `argv[0]` = "copy", `argv[1]` = "C:/User/archivo.ext" y `argv[2]` = "C:/archivo.ext" (sin las comillas).

## INPUT

El hola mundo que hicimos es tan solo un programa que escribe a pantalla. Es decir, produce una salida(output). Sin embargo este no hace nada más. Además no recibe ninguna entrada(input) del usuario. Los típicos programas realmente muestran salidas que dependen de entradas dadas por el usuario del programa.

## Variable

Para poder leer algo, necesitamos un lugar donde poner lo leído, i.e. necesitamos un lugar en la memoria de la maquina donde podamos guardar esta información. A este “lugar” lo llamaremos **objeto**. Un objeto es una región de memoria que tendrá un **tipo** que especifica que clase de información esta siendo colocada en el. Este objeto es llamado **variable**. Será en estas variables donde guardaremos información en nuestros programas.

bool x	x es a Booleano (valor true and false).
char x	x is a character (usually 8 bits).
short x	x is a short int (usually 16 bits).
int x	x is the default integer type.
float x	x is a floating-point number.
double x	x is a double-precision floating-point number.
void *p	p is a pointer to raw memory.
const T x	x is a constant (immutable) version of T.
long T x	x is a long T.
unsigned T x	x is an unsigned T.
signed T x	x is a signed T.

specifier	Output	Example
c	Character	a
d or i	Signed decimal integer	392
e	Scientific notation using e character	3.9265e+2
E	Scientific notation using E character	3.9265E+2
f	Decimal floating point	392.65
g	Use the shorter of %e or %f	392.65
G	Use the shorter of %E or %f	392.65
o	Signed octal	610

specifier	Output	Example
s	String of characters	sample
u	Unsigned decimal integer	7235
x	Unsigned hexadecimal integer	7fa
X	Unsigned hexadecimal integer	7FA
p	Pointer address	B800:0000
n	Nothing printed.	



## Ejemplo

```
#include <stdio.h>
int main(){
    printf ("Characters: %c %c \n", 'a', 65);
    printf ("Decimals: %d %ld\n", 1977, 650000L);
    printf ("Preceding with blanks: %10d \n", 1977);
    printf ("Preceding with zeros: %010d \n", 1977);
    printf ("Some different radixes: %d \n", 100);
    printf ("Some different radixes: %x \n", 100);
    printf ("Some different radixes: %o \n", 100);
    printf ("Some different radixes: %#x \n", 100);
    printf ("Some different radixes: %#o \n", 100);
    printf ("floats:%4.2f %+.0e %E\n",3.1416,3.1416,3.1416);
    printf ("%s \n", "A string");
    return 0;
}
```

¿Como hago que el usuario entre datos?

```
int scanf ( const char * format, ... );
```

## Ejemplo

```
#include <stdio.h>

int main (){
    char str[80];
    int i;
    printf ("Enter your family name: ");
    scanf ("%s",str);
    printf ("Enter your age: ");
    scanf ("%d",&i);
    printf ("Mr. %s , %d years old.\n",str,i);
    printf ("Enter a hexadecimal number: ");
    scanf ("%x",&i);
    printf ("You have entered %#x (%d).\n",i,i);
    return 0;
}
```

## Como hacer comentarios

```
// esto comenta una linea.  
/*  
    esto comenta  
    un bloque de instrucciones  
*/
```

 Programming Principles and Practice Using C++, Bjarne Stroustrup.

 <http://www.codeblocks.org>

 <http://www.wxwidgets.org>