****

**Input/ Output**

C language has the ability to handle input / output operations, which are defined in the standard I/O system for C programs.

Input/output features of C are printf, scanf, getchar and putchar functions.

A program using these functions must be included in the standard header file <stdio.h>

in the directive

**The printf ( ) function**

**printf(control string, arg1, arg2,……….. . . );**

1. The argument or arg1, arg2 are the individual output data items.
2. The arguments can be variables, consonants or array names
3. printf don’t have memory addresses
4. Which is why they are not preceded by ampersands ( &)
5. printf moves data from the computer’s memory to the standard output devices.

**Format Descriptors**

**Format strings define how input /output data is represented and interpreted internally**

**A control sequence for every control string in each argument**

|  |  |
| --- | --- |
| **Conversion control character** | **Meaning** |
| **%d** | **Print and scan as signed decimal integer** |
| **%u** | **Print and scan as unsigned decimal integer** |
| **%s** | **Print and scan as a string** |
| **%f** | **Print and scan as floating point number** |
| **%e** | **Print and scan as scientific notation** |
| **%c** | **Print and scan as a character** |

**Escape Sequences**

When a backslash(\) is used in front of a selected group of characters, the compiler tells the computer to escape from the way these characters are normally interpreted.

Backslash and specific characters are called as Escape sequences

|  |  |
| --- | --- |
| Escape Sequence | Meaning |
| \n | Newline |
| \t | Horizontal tab |
| \v | Vertical tab |
| \a | Beep the speaker |

Example 1:

Printf(“%c”, \’m’ \);

**Output is m**

Example 2:

#include<stdio.h>

Int no =10;

Printf(“%d\n”, 9 \* no);

**Output is 90**

**The scanf( ) function**

scanf(control string, arg1, arg2,….. . . );

scanf("%d”, &no);

1. scanf function gets the value from the user and stores the value in the memory of the

 variable no which is represented as &no

2. &no refer to the address of the variable.

3. reads one data item from the input corresponding to each argument other than the control string

4. It returns total number of functions successfully read

5. It skips newline, whitespaces.

6. It returns EOF when the end of the input is reached.

**Example:**

**#include <stdio.h>**

 **main( )**

 **{**

 **Int no;**

 **scanf("%d”, &no);**

**}**

**The getchar( ) function**

It’s a part of I/O library of C language

It returns a single character from a standard input device

character variable = getchar();

-------------------------------------------------------------------------------------------------------------------------

\*\*When It is equated to an int value, it returns the ASCII value of the variable.

Example:

#Include<stdio.h>

 main()

 {

 int ch;

 ch =getchar ( );

 printf(“%d”,ch);

}

----------------------------------------------------------------------------------------------------------------

\*\*When It is equated to an character, it returns the character typed in.

Getchar( ) function returns the ASCII value of the character

Example:

#Include<stdio.h>

 main ()

 {

 char ch;

 ch =getchar ( );

 printf(“%c”,ch);

}

**The putchar ( ) function**

It is one of function from standard C language I/O library.

It prints the character back on the screen

Character is enclosed in the parentheses

 putchar(character variable)

Example:

#include<stdio.h>

 main()

{

int ch ;

 ch= getchar(); /\*returns the ASCII value of the character \*/

 puts( ch) /\*prints the character \*/

}