Chapter 4

MANAGING INPUT OUTPUT OPERATIONS

Reading a Character:-

Reading a single character can be done by using the function getchar(). this can also be done with scanf().

Syntax:

```
ch =    getchar() ;
```

Input /Output Statements:-

To perform the basic i/o functions C provides the library of functions. This library is called stdio.h

Ex:

scanf() , printf(), getchar(), getch(), putchar(), gets(), puts(), etc.

There are two types of I/O statements. They are,

- **Formatted I/O Statement**
- **Unformatted I/O statement**

1) **Formatted I/O St:-**

   This enables the user to specify the type of the data & the way in which it should be read in or written out.

   Ex:-scanf(), printf()

2) **Unformatted I/O St:-**

   This do not specify the type of data & the way it should read in or written out.

   Ex:- getchar(), gets(), putchar(), puts(), etc

Scanf():-

Syntax:

```
scanf("Control String",address_list);
```

where,

"control string" is a sequence of one or more character groups. Each character group is a combination of % symbol and one of the conversion characters. The control string specifies the type of the values which are to be supplied to the variables.

"address list" are address of memory locations where the values of input variable should be stored.

Character Group	Meaning
%c	read a single character
%d	read a decimal integer
%f	read a floating point value
%u	read a unsigned value

Interger Input:-

```
Ex:-    num =386;
        scanf("%3d",&num) ;
```

where,

3 is the field widthof the input number.

Note:

No space or any other characters is used after te last character group.

```
Ex:     scanf("%d %d %d    ",&p,&q,&r);
```

In this case ,the programmer has to input one char extra to end the input operation.

Similarly Formatted output:

Using printf()

```
printf("%05d",678);
```

0	0	6	7	8

gets() :-

syntax:

```
gets(string);
```

unformatted output:-

```
putchar(char_variable);
puts(string);
```