* + % operator is not to be used in float and string.

**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 statements
  + Unformatted I/O statements
* **Formatted I/O statements:**

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

Ex : scanf(), printf().

* **Unformatted I/O statements:**

This do not specify the type of data and 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"

|  |  |
| --- | --- |
| Character Group |  |
| %c | Read a single char |
| %d | Read a decimal integer |
| %f | Read a floating point value |
| %u | Read a unsigned value |

* "0d" implies that when we want the format filled with the 0 instead of the empty blocks then we can use "0" so that the empty will be filled with 0 and remaining will be filled with the given values.
* In printf you will be de referencing the value so that it can be stored.
* To remove buffer we use "fflush".

Error :

"Stack smashing detected" : terminated

>> This error occurs because c language does not have a garbage collector and all the memory stored up while executing the code is more.

Addresses of

i : 1440203472

f : 1440203476

ch : 1440203471

str1 : 1440203488

d : 1440203480

These are known as base addresses of any variables or where the addresses are stored at the starting.