**Day 7**

**Terenary operator :**

We use the ternary operator in C to run one code when the condition is true and another code when the condition is false. For example,

(age >= 18) ? printf("Can Vote") : printf("Cannot Vote");

Here, when the age is greater than or equal to 18, Can Vote is printed. Otherwise, Cannot Vote is printed.

The syntax of ternary operator is:

testCondition ? expression1 : expression 2;

The testCondition is a boolean expression that results in either **true** or **false**. If the condition is

* true - **expression1** (before the colon) is executed
* false - **expression2** (after the colon) is executed

The ternary operator takes 3 operands (condition, expression1 and expression2). Hence, the name **ternary operator**.

**Managing input output operator :**

Reading a character

getchar()

ch = getchar();

2 types of input output statements

1. formatted I/O statement
2. Unformatted I/O statement

**formatted I/O statement:**

printf()

scanf()

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

**Unformatted I/O statement :**

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

Scanf(“control string”,address\_list);

%u 🡪 unsigned value it will be as unsigned integer

%lf 🡪 double

Ex:

#include<stdio.h>

int main()

{

int num ;

scanf(“%d”,&num);

printf(“%3d”,num);

return 0;

}

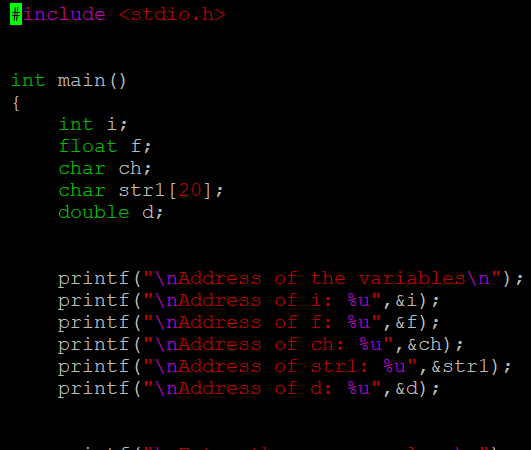
In printf we have to dereference the value while scanning a string we should not specify the address as scanf(“%s”,str);

**Flush** is used to clear the temporary buffer

Garbage collector is the major difference between java and c++

**Error :** Stack smashing detected terminated aborted (core dumped)

This occurs to clean the memory when we use out of bounds.



In the above code the address of variables is printed

i: 1440203472

f1:440203476

ch: 1440203471

str1: 1440203488

d: 1440203480

this is called as base address of variable.

* In scanf there should not be any spaces for the starting and ending of a format specifiers " %d " or " %d" or "%d ".

if the user required spaces the space can be given in between like "%d %d" (or) "%d%d"

\t and \n can not be given in scanf because it takes as an character .

* to convert integer to string or alpha itoa is used or sprintf is used

to convert from alpha to integer atoi is used

to use this <stdlib.h> is included.

unformatted i/o;

getc,putc======> read/write single char =>stream

getchar,put char ======>read/write single char=> std i/o

gets,puts=========> read/write for a string=>stream

getch()=>DOS platform =>not echo read char to the sreen.