**DAY07**

Managing input output operations:

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

Syntax: ch = getchar ()

Input and output statements: To perform the basic i/o function c provide the library of functions. This library is called stdio.h.

ex : scanf (), printf (), getchar (), putchar (), gets()

There are two types of I/O statements:

1)Formatted I/O stat: This enables the user to specify the type of data & the way in which it should be read in or written out.

Ex: scanf () , printf ()

2)UnFormatted I/O stat: This do not specify the type of data & the way it should read in or written out.

Ex: getchar (), gets(), putchar ()

Scanf():

Syntax: scanf(“control string”, address\_list);

Address\_list: Are address of memory location where the values of input.

Character Group Meaning

%c Read of single character

%d Read of Decimal character

%f Read of float

%u Read of unsigned

Integer Input:

Ex : num =386;

Scanf(“%03d”,&num);-> Here 3 is the field width of the input number.Other will in buffer

Similarly Formatted output: using printf()

Printf(“%05d”,678)-> Here 3 is the field width and remaining 2 will fill with 0’s

=> #include<stdio.h>

2 int main()

3 {

4 int i;

5 float f;

6 char ch;

7 char str1[20];

8 double d;

9 printf("\nEnter the proper values\n");

10 printf("\nEnter the integer value :\n");

11 scanf("%d",&i);

12 printf("\nEnter the float value:\n");

13 scanf("%f",&f);

14 printf("\nEnter the double value :\n");

15 scanf("%lf",&d);

16 printf("\nEnter the string:\n");

17 scanf("%s",str1);

18 printf("\nEnter the single character value:\n");

19 if (scanf("%c",&ch)>0)

20 printf("successfully scanned");

21 else

22 printf("not scanned");

23

24

25 printf("\nInteger value:%d\n",i);

26 printf("\nfloat value:%f\n",f);

27 printf("\ndouble value:%lf\n",d);

28 printf("\nstring value:%s\n",str1);

29 printf("\ncharacter value:%c\n",ch);

30 printf("\n\n");

31 return 0;

=>

#include<stdio.h>

2 int main()

3 {

4 int i;

5 float f;

6 char ch;

7 char str1[20];

8 double d;

9 printf("\nEnter the proper values\n");

10 printf("\nEnter a single character value\n");

11 scanf("%c",&ch);

12 printf("\nEnter the integer value :\n");

13 scanf("%d",&i);

14 printf("\nEnter the float value:\n");

15 scanf("%f",&f);

16 printf("\nEnter the double value :\n");

17 scanf("%lf",&d);

18 printf("\nEnter the string:\n");

19 scanf("%s",str1);

20

21

22

23 printf("\nInteger value:%d\n",i);

24 printf("\nfloat value:%f\n",f);

25 printf("\ndouble value:%lf\n",d);

26 printf("\nstring value:%s\n",str1);

27 printf("\ncharacter value:%c\n",ch);

28 printf("\n\n");

29 return 0;

30 }

\*while scanning a string, we should not mention address.

\*To avoid temporary buffer, we can declare single character within initial stage (or)fflush :it does not work on linux.

=>Assignment:

#include<stdio.h>

2 int main()

3 {

/\* These are static variables which are stored in stack memory\*/

4 int id;

5 int phno;

6 float salary;

7 char name[20];

8 char adress[20];

9 char gender;

/\*To know the address of these static variables.These are called as base address of variables.

10 printf("\naddress of id:%u",&id);

11 printf("\nEnter the Employess Details\n");

12 printf("\nEnter ID : ");

13 scanf("%d",&id);

14 printf("\nEnter Phnone number: ");

15 scanf("%d",&phno);

16 printf("\nEnter salary : ");

17 scanf("%f",&salary);

18 printf("\nEnter Name : ");

19 scanf("%s",name);

20 printf("\nEnter adress : ");

21 scanf(" %s",adress);

22 printf("\nEnter gender:");

23 scanf(" %c",&gender);

24

25 printf("\nSINO | ID | phoneno | salary | name | adress | gender\n");

26 printf("\n%03d |%d |%d|%f|%s|%s|%c",1,id,phno,salary,name,adress,gender);

27 return 0;

28 }

=> #include<stdio.h>

2 int main()

3 {

4 int i,j;

5 scanf("%d %d",&i,&j);

6 printf("\ni=%d\tj=%d",i,j);

7 return 0;

unformatted i/o:

\*getc, putc => read/write single char -> stream

\*getchar, putchar => read/write single char =>std i/o

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

#include<stdio.h>

9 int main()

10 {

11 char ch;

12 printf("\nEnter a character: ");

14 ch=getchar(); =>To get input from user

15 printf("\nread char: ");

16 putchar(ch); =>To Display that input

17 putchar('A');

18 putchar(65);

19

20 ch = fgetc(stdin);

21 printf("\nRead char:");

22 putc(ch,stdout);

23 printf("\n\n");

24

25 return 0;

26 }

Decision making and Branching: \*When we have different conditions to check we use branching.

1)If: if(cond) : if it is single statement we don’t use braces.

{

If block if stats

}

2)If else:

1 #include<stdio.h>

2 int main()

3 {

4 int age;

5 scanf("%d",&age);

6 if(age>=18)

7 {

8 printf("\nyou are eligible to vote");

9 }

10 else

11 {

12 printf("\nyou are not eligible to vote");

13 }

14 printf("\nIndian citizen\n");

15

16 return 0;

17 }

18

3)Nested if else:

If(cond1)

{

If(cond2)

{

}

Else

{

}

}

Else

{

If(cond3)

{

}

Else

{

}

}