**DAY-11**

**STRUCTURES**

/\*

**user defined DT**

UDT:

structures

unions

enums

struct, union, enum

**Code:**

struct tagName

{

members of structure

};

struct Square

{

int len;

int breadth;

};

struct Chair

{

int noLegs;

char make[20];

char material[20];

char clr[20];

float price;

char DOM[20];

char placeManu[20];

};

=====================================================

int a1,a2,;

struct Square s1,s2,s3,sq[10];

struct Square

{

int len;

int breadth;

}s4,s5,s6,s[10];

typdef struct Square SQR;

SQR s7,s8,s9;

typdef struct Square

{

int len;

int breadth;

}SQR1;

SQR1 s1;

s1.len

1) . => static var => nameVar.memberName

2) -> => ptr var => nameVar->memberName

SQR1 \*ptr;

ptr->len;

void func(struct Square s)

{

}

\*/

--------🡪

char a='D'

char Name[5]={'H','e','l','l','0'};

int arr[5] ={1,2,3,4,5};

for(i=0;i<5;i++)

printf("%d",arr[i]);

for(i=0;i<5;i++)

printf("%c",Name[i]);

char Name[6] = {'H','e','l','l','0','\0'};

char Name[6] = "Hello";

char Greet[] = "Hello World";

char Str1[21];

char Name[][20] = {"Bhima","shankar","Amit", "manju", "Meena"};

// there are 10 names each of max cap of 50 chars

char Names[10][50];

**Code1:**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int \*ptr=NULL;

ptr = (int \*)malloc(3\*sizeof(int));

free(ptr);

// free(ptr);

printf("\n\n");

return 0;

}

**Code2:**

#include <stdio.h>

int main()

{

char Name[5];

int i;

for(i=0;i<20;i++)

scanf("%c",&Name[i]);

Name[5-1] ='\0';

for(i=0;i<20;i++)

printf("\n%c=%d",Name[i],Name[i]);

printf("\n\n");

printf("\n%s\n",Name);

puts(Name);

printf("\n\n");

return 0;

}