Structures – A simple variable can store a single value only. An array can store multiple values but with similar data type only. However in many real life applications, we need to store multiple values with disimilar data types also. In such cases, we use “structures”. For eg. if we want to store records of students i.e. their rollnos, names & percent OR records of say employees i.e. their empids, names, age, salary, etc.

Definition – A structure is a collection of similar and/or dissimilar datatypes.

Syntax to declare a structure

struct structure\_name // any name can be assigned to structure like var\_name, function\_name, etc.

{

basic\_datatype1 var1, var2, var3, ……;

basic\_datatypen vara, varb, varc, ……; Elements/Members of structure

……..

};

The above is only the definition of structure with no memory allocated to it. So to actually store data, we need to define vars or arrays of structure.

struct structure\_name var1, var2, array\_name[10];

For e.g. to store data of employee

struct emp

{

int empid, age;

char name[10];

float salary;

};

struct emp e1, e2;

printf(“Size of e1 is %d”,sizeof(e1)); // 22 bytes

e1 e2

empid - ?

age - ?

name – ?

salary - ?

empid - 123

age - 30

name – satish

salary – 25600.36

To access the members of structure, we use . (dot) syntax

struct\_var\_name.member\_name

e1.empid = 123;

e1.age = 30;

strcpy(e1.name,”satish”); // e1.name = “satish”; error

e1.salary = 25600.35

To store multiple records, of employees, we will use “array of structures”

struct emp recs[5];

0 1 2 3 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 123  35  rajesh  10000.00 | 234  23  mahesh  25000.35 | 345  29  yatin  30000.00 | ….. | ….. |

recs[5] 🡪 empid

age

name

salary

To display record of 3rd employee from array recs[ ]

printf(“%d %s %d %.2f”,recs[2].empid,recs[2].name,recs[2].age,recs[2].salary);

**Note –** Structures do not have their own “format specifiers”. Hence, we cannot print the data of structure directly. So we need to access the members using their format specifiers.