Date No.

Structures in C:

Structures in C is the user-defined datatype which is the collection of one or more variables of different datatypes grouped together under a single name.

(*) Advantages:

i) It makes the application program easier to read and understand.
ii) Applications are less likely contain logic error.
iii) Errors can be easily debugged.

(*) Declaration of structure:

Syntax: struct structuretag
{ int rollno; member 1;
char name; member 2;
:
};

Eg: struct student
{ int rollno;
char name[20];
float cgpa;
};

→ two members of a structure can't be same.

(*) Declaration of Objects of structure:

The variables used for structure datatype is called objects of structure. It is called structure variables.

Syntax: ~~structure~~ datatype variablename;

Eg:
```
struct student
{
 int roll;
 float cgpa;
 char name[20];
};
void main()
{
 struct student s;
}
```
OR
```
struct student
{
 int roll;
 char name[20];
 float cgpa;
} s;
```

(*) ~~Initigation~~ Initialization and Accessing Structure:

→ Initialization cannot be done within structure.

~~Syntax: struct studer~~

At compile time;

```
struct student s={1, "A", 3.7};
```

At run time;

```
struct student s;
scanf("%d", &s.rollno);
scanf(" %[^\n]", s.name);
scanf(" %f", &s.cgpa);
```

To initialize more than two variables.

Eg:
```
struct student
 { int rollno; char name[20]; float cgpa } s1, s2, s3;
```

or,
```
struct student s1, s2, s3;
```

(*): Accessing: (structure objectname . membername)

Eg:
```
printf("%d", s1.rollno);
printf("%s", s1.name);
printf("%f", s1.marks);
```

⊛ Rules of Initisation:

i) Order of members in initialization is the same as the order of members in structure declaration.

ii) If partial initialization is done,
Eg: struct student s1={1};
then, rollno contains 1 but the name is null and float marks has 0.

Note: i) Assigning one structure variable to another is allowed ie, s1=s2.
ii) The comparison of two structure variables is not possible.
but the individual members can be compared
Eg: if (s1.roll < s2.roll)

⊛ Array of Structure:

Array of structure is used to store more student data.

Eg: struct student
{ int roll; char name[20]; float cgpa; } s[60]

Accessing and initialization of array of structure is done using loops.

Initisation	Accessing
Eg: for (i=0; i<n; i++)	for (i=0; i<n; i++)
{	{
scanf ("%d", &s[i].roll);	printf ("%d", &s[i].roll);
scanf ("%[^\n]", s[i].name);	printf ("%s", s[i].name);
scanf ("%f", &s[i].cgpa);	printf ("%f", s[i].cgpa);
}	}

⊛ Pointer to structure:

```
struct student
{ int rollno; char name[25]; float cgpa; };
void main()
{ struct student s;
  struct student *ptr;
  ptr = &s;

scanf("%d", &s.rollno);
scanf("%[^\n]", s.name);
scanf("%f", &cgpa);
```

Displaying using pointers;
It can be done in three ways.

- printf ("Rollno = %d", s.rollno); ——— (i)
- printf ("~~Name = %s~~" "Rollno = %d", ptr -> rollno); — (ii)
- printf ("Rollno = %d", (*ptr).rollno); —— (iii)