STRUCTUREAND UNION

Q. What is structure? How a structure is declared? How does a structure differ from an array? Marks:3 Exam-ACCE-2014

Ans:

Structure: A structure is a collection of variables under a single name. These variables can be of different types, and each has a name which is used to select it from the structure. A structure is a convenient way of grouping several pieces of related information together.

Q. How do declare a structure variable? Exam:ICE-2013,CSE,APPE

Declaring Structure Variable in C

In C we can group some of the user defined or primitive data types together and form another compact way of storing complicated information is called as Structure. Let us see how to declare structure in c programming language

syntax Of Structure in C Programming :

```
struct tag
{
  data_type1 member1;
  data_type2 member2;
  data_type3 member3;
};
```

Some Important Points Regarding Structure in C Programming :

1. **Closing brace of structure type declaration must be followed by semicolon.**

```
struct date

  int date
  char month
  int year
```

2. **Don't forgot to use 'struct' keyword**

```
struct date
```

3. **Memory will not be allocated just by Creating instance or by declaring structure**

Memory will not be allocated just by writing following declaration –

```
struct date

  int date
  char month
  int year
```

Memory will be allocated on defining it i.e –

```
date d1
```

Generally Structures are written in Global Declaration Section , But they can be written inside main.

```
int main

  struct student_database

    char name
    int roll
    int marks
```

```
stud1

return
```

5.It is not necessary to initialize all members of structure.

```
struct student_database
;
        char name
        int roll
        int marks
      stud1    "        "
```

6.For Larger program , Structures may be embedded in separate header file.

Different Ways of Declaring Structure Variable :

Way 1 : Immediately after Structure Template

```
struct date
{
    int date;
    char month[20];
    int year;
}today;
```

// 'today' is name of Structure variable

Way 2 : Declare Variables using struct Keyword

```
struct date
{
   int date;
   char month[20];
   int year;
};
struct date today;
```

where "date" is name of structure and "today" is name of variable.

Way 3 : Declaring Multiple Structure Variables

```
struct Book
{
   int pages;
   char name[20];
   int year;
}book1,book2,book3;
```

We can d eclare multiple variables separated by comma directly after closing curly.

Q. Array of .structure? Exam:CSE-2011,APPE,ACCE-2011

The first parameter to REPLICATE is a reference to the structure of each element. Using the example in Examples of Structure References and assuming the STAR structure has been defined, an array containing 100 elements of the structure is created with the following statement: cat = REPLICATE({star}, 100

Q. what is the difference between array and structure? Exam:ACCE-2011,ICE-2013,CSE

Ans:

The following are the differences between structures and arrays:

Array elements are homogeneous. Structure elements are of different data type.

Array allocates static memory and uses index / subscript for accessing elements of the array.

Structures allocate dynamic memory and uses (.) operator for accessing the member of a structure.

Array is a pointer to the first element of it. Structure is not a pointer

Array element access takes less time in comparison with structures.

Q. How do you use structure as function parameter?
Ans:

Passing structure to function in C:

- A structure can be passed to any function from main function or from any sub function.
- Structure definition will be available within the function only.
- It won't be available to other functions unless it is passed to those functions by value or by address(reference).

It can be done in below 3 ways.

1. Passing structure to a function by value
2. Passing structure to a function by address(reference)
3. No need to pass a structure – Declare structure variable as globa

Example program – passing structure to function in C by value

In this program, the whole structure is passed to another function by value. It means the whole structure is passed to another function with all members and their values. So, this structure can be accessed from called function.

```
#include<stdio.h>
struct student{
 int id;
 char name[20];
 float percentage;
};
int func(struct student record);
int main()
 {
    struct student record;
    record.id=1;
    strcpy(record.name,"Raju");
    record.percentage=86.5;
    func(record);
    return 0;
 }
 int func(struct student record)
 {
    printf("Id is: %d\n",record.id);
    printf("Name is:%d\n",record.name);
    printff("percentage is:%f \n",record.percentage);
 }
```

Q.what do you mean by structure?
Ans:

Defining a Structure

To define a structure, you must use the **struct** statement. The **struct** statement defines a new data type, witl more than one member for your program.

The format of the struct statement is this:

struct [structure tag]

```
{
member definition;
member definition;
...
member definition;
} [one or more structure variables];
struct Books
{
}
```

Q. How many ways to create custom data types?
Ans:

1. Alias types using Simulink.AliasType
2. Custom numeric types using Simulink.NumericType
3. Structure types using Simulink.Bus
4. You can also define a Custom Storage.Class if you want to introduce a new storage class specification.

Q. How to assign values to struct members?
Ans:

There are three ways –

1) Using Dot(.) operator

```
var_name.memeber_name = value;
```

2) Whole structure's variables assignment in one go –

```
struct struct_name   var_name =
{value for memeber1, value for memeber2 ...so on for all the members}
```

3) Designated initializers –

Example:

```
#include <stdio.h>
struct StudentData{
  char *stu_name;
  int stu_id;
  int stu_age;
}

int main()
{
   struct StudentData  student;
   student = {"Chaitanya", 1234, 25};

   printf("Student Name is: %s", student.stu_name);
   return 0;
}
```

There are two things to notice in above example –
1) Both the below code snippets are same.

```
student = {"Chaitanya", 1234, 25};
```

OR(both are same!!)

```
student.stu_name = "Chaitanya";
student.stu_id = 1234;
student.stu_age = 25;
```

2) Dot (.) operator can be used to assign values to individual members of structure or can be used to access values stored in structure's individual members.

Q. what is union?What is its general form? Give an example showing how different types of data share the same memory location? Marks:3 Exam-ICE-2013,CSE-ACCE-2014

Unions: Aunion is a special data type available in C that enables you to store different data types in the same memory location..

Defining a Union

A *union*is an object similar to a structure except that all of its members start at the same location in memory.

- A union variable can represent the value of only one of its members at a time
- So an union is a variable which may hold (at different times) objects of different sizes and types

```
general form of union:
union [union tag]
{
member definition;
member definition;
...
member definition;
} [one or more union variables];
```

The union tag is optional and each member definition is a normal variable definition, such as int i; or float f; or any other valid variable definition. At the end of the union's definition, before the final semicolon

Example:

```
union number
{
short shortnumber;
long longnumber;
double floatnumber;
} anumber
```

It defines a union called **number** and an instance of it called anumber.

Members can be accessed in the following way: printf("%d\n",anumber.longnumber)

When C compiler is allocating memory for unions, it will always reserve enough room for the largest member.

In the above example this is 8 bytes for the double.

```
union u_t
{ char a;
short b;
intc;
};
union u_tx;
x.a= 'B';printf("%c\n", x.a);
```

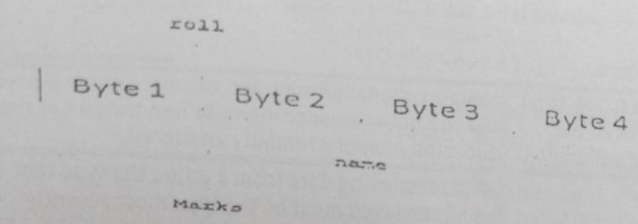
Q.Explain how memory is allocated for a union in C? Exam:2011ICE-2013

```
int roll
char name
int marks
```

We have collected three variables of different data type under same name together.

All Union Members Occupy Same Memory Area

February 1, 2016



For the union maximum memory allocated will be equal to the data member with maximum size. In the example character array 'name' have maximum size thus maximum memory of the union will be 4 Bytes.

Maximum Memory of Union Maximum Memory of Union Data Member

Only one Member will be active at a time.

Suppose we are accessing one of the data member of union then we cannot access other data member since we can access single data member of union because
By Using Union we can Save Lot of Valuable Space
Simple Example :

```
union u

char a
int b
```

Q.Distinguish between structure and union? Marks:1.50 Exam-ACCE-2013
Ans:

Difference between structure and union in C	
structure	**union**
Keyword struct defines a structure.	Keyword union defines a union.
Example structure declaration: struct s_tag { int ival; float fval; char *cptr; }s;	Example union declaration: union u_tag { int ival; float fval; char *cptr; }u;
Within a structure all members gets memory allocated and members have addresses that increase as the declarators are read left-to-right. That is, the members of a structure all begin at different offsets from the base of the structure. The offset of a particular member corresponds to the order of its	For a union compiler allocates the memory for the largest of all members and in a union all members have offset zero from the base, the container is big enough to hold the WIDEST member, and the alignment is appropriate for all of the types in the union. When the storage space allocated to the union

declaration; the first member is at offset 0. The total size of a structure is the sum of the size of all the members or more because of appropriate alignment.	contains a smaller member, the extra space betwee the end of the smaller member and the end of the allocated memory remains unaltered.
Within a structure all members gets memory allocated; therefore any member can be retrieved at any time.	While retrieving data from a union the type that is being retrieved must be the type most recently stored. It is the programmer's responsibility to keep track of which type is currently stored in a union; the results are implementation-dependent if something is stored as one type and extracted as another.
One or more members of a structure can be initialized at once.	A union may only be initialized with a value of the type of its first member; thus union u described above (during example declaration) can only be initialized with an integer value.

Q.Write a program that read name and roll and print?

```
#include <stdio.h>
struct student{
   char name[50];
   int roll;
};
int main(){
   struct student s1;
   printf("Enter student's name: ");
   scanf("%s",&s1.name);
   printf("Enter roll number:");
   scanf("%d",&s1.roll);
   printf("Output\nName: %s",s1.name);
   printf("\nRoll: %d",s1.roll);
   return 0;
 }
```

Q.Discuss Characteristics of Structure?

Ans:

1) A struct is a collection of other data types.
2) The length of a struct is fixed based on the data type of the members.
3) A struct can be nested.
4) typedef creates a type name for a struct and makes the code easier and readable.
5) struct fields can be accessed using Dot(.) operator.

Q. Write a program using structure that store 100 student's name roll and total marks?

Ans:

```
#include <stdio.h>
#include <conio.h>
int main()
   {
   struct student
        {
        int rollno;
        char name[25];
        int totalmark;
        }stud[100];
   int n,i;
```

```
clrscr();
printf("Enter total number of students\n\n");
scanf("%d",&n);
for(i=0;i<n;i++)
      {
      printf("Enter details of %d-th student\n",i+1);
      printf("Name:\n");
      scanf("%s",&stud[i].name);
      printf("Roll number:\n");
      scanf("%d",&stud[i].rollno);
      printf("Total mark:\n");
      scanf("%d",&stud[i].totalmark);
      }
printf("STUDENTS DETAILS:\n");
for(i=0;i<n;i++)
      {
      printf("\nRoll number:%d\n",stud[i].rollno);
      printf("Name:%s\n",stud[i].name);
      printf("Totel mark:%d\n",stud[i].totalmark);
      }
getch();
return 0;
}
```