

PROGRAMMING



FOR PROBLEM SOLVING

{ C LANGUAGE }



Module :- 8

Structure And Union

Start



Module 8 :- Structure And Union

- Structures, accessing structure elements
- Way of storage of structure element
- Defining structures and array of structures
- Basic definition of union
- Comparison between structure and union with example

Structure

- It is a collection of data of different data type
- It is a user defined data type
- Data can of int , char , float , double etc data type
- We can access the member of structure by making the variable of structure
- Struct keyword is used to create a structure

Syntax

```
struct structure_name  
{  
    data_type variable 1;  
    data_type variable 2;  
    .....  
    .....  
    data_type variable n;  
};
```



example

```
Struct student  
{  
Char name [200];  
Int rollno;  
Float marks;  
};
```

- Here student is the name of structure
- Struct is a keyword



Declaration of structure variable methode 1

```
Struct student
```

```
{
```

```
Char name [200];
```

```
Int rollno;
```

```
Float marks;
```

```
};
```

Here student1 is the variable of structure

```
Int main ()
```

```
{
```

```
Struct student student1;
```

```
Return 0;
```

```
}
```



Declaration of structure variable methode 2

```
Struct student
{
Char name [200];
Int rollno;
Float marks;
}student1;
Int main ()
{

Return 0;
}
```



Accessing the data members of structure

The data member of structure can be accessed as

`Structure_variable.data_member`

For example

If we want to access the rollno of student then we can write as

`Student1.rollno`

Example -> write a program to store and display the student name , rollno and marks



```
#include<stdio.h>
#include<string.h>
struct student
{
    char name[200];
    int rollno;
    float marks;
};
int main()
{
    struct student student1;//Declaring structure variable
    strcpy(student1.name,"Lucky");
    student1.rollno=201;
    student1.marks=85.9;
    printf("Student Name =%s\n",student1.name);
    printf("Student Rollno=%d\n",student1.rollno);
    printf("Student Marks=%f\n",student1.marks);
}
```

```
### Output ###
Student Name=Lucky
Student Rollno=201
Student Marks=85.9
```

example



1. Input name roll no. and marks of student and print using structure
2. Addition program using structure
3. User input in structure
4. Subtraction program using structure
5. Multiplication program using structure
6. Division program using structure

Union

- It is a collection of data of different data type
- It is a user defined data type
- Data can of int , char , float , double etc data type
- We can access the member of union by making the variable of structure
- union keyword is used to create a union
- **Note** -> union does not support multiple value simultaneously it can store only one value at a time

Syntax



```
union union_name
{
data_type variable 1;
data_type variable 2;
.....
.....
data_type variable n;
};
```

example



```
Union student
{
Char name [200];
Int rollno;
Float marks;
};
```

- Here student is the name of union
- union is a keyword



Declaration of union variable methode 1

```
union student
{
Char name [200];
Int rollno;
Float marks;
};
Int main ()
{
union student student1;

Return 0;
}
```

Here student1 is the variable of union

Declaration of structure variable methode 2



```
union student
{
Char name [200];
Int rollno;
Float marks;
}student1;
Int main ()
{

Return 0;
}
```



Accessing the data members of union

The data member of union can be accessed as

`union_variable.data_member`

For example

If we want to access the rollno of student then we can write as

`Student1.rollno`

Example -> write a program to store and display the student name , rollno and marks



Note -> but union will show only last value correct because it can store only single value at a time
iam writing this program here so that you can understand difference b/w structure and union

```
#include<stdio.h>
#include<string.h>
union student
{
    char name[200];
    int rollno;
    float marks;
};
int main()
{
    union student student1;//Declaring union variable
    strcpy(student1.name,"Lucky");
    student1.rollno=201;
    student1.marks=85.9;
    printf("Student Name =%s\n",student1.name);
    printf("Student Rollno=%d\n",student1.rollno);
    printf("Student Marks=%f\n",student1.marks);
}
```

Output

Student Name=gabrage value
Student Rollno=gabrage value
Student Marks=85.9



Enumeration

- It is collection of named integer constant
- It is user defined data type
- Enum keyword is used to create a enumeration

- `enum enum_name { value1,value2,.....valueN };`



Example

```
enum week{Sunday,Monday,Tuesday,Wednesday,thrusday,Friday,Saturday};
```

- Enum is a keyword
- Week is the name of enum and it is a user defined data type
- Sunday,Monday,Tuesday,Wednesday,thrusday,Friday,Saturday are the value of enum



Default numeric value of the member of enum is given below

- Default value of Sunday is 0
- Default value of Monday is 1
- Default value of Tuesday is 2
- Default value of Wednesday is 3
- Default value of Thursday is 4
- Default value of Friday is 5
- Default value of Saturday is 6



```
#include<stdio.h>
enum week {sunday,monday,tuesday,wednesday,thrusday,friday,saturday};
int main()
{
enum week obj;
obj=wednesday;
printf("Value of wednesday=%d",obj);
}
```

Output ###Value of wednesday=3



We can also change the default value of member of enum

```
#include<stdio.h>
enum week{sunday=20,monday=50,tuesday=18,wednesday=95,thrusday=84,friday=60,saturday=55};
int main()
{
enum week obj;
obj=wednesday;
printf("Value of wednesday=%d",obj);
}
```

Output ###Value of wednesday=95



Thank You !!

Dhanybad !!

Shukriya !!