

Structure and Unions

Week 9

What is a Structure?

- Structure is the collection of variables of different types under a **single name** for better handling.
- **For example:** You want to store the information about person about his/her name, citizenship number and salary.
- Keyword struct is used for creating a structure.

Using Structures

- Define the structure.
- Declare/Initialize instances of the structure.
- Access members of an instance of the structure.

Syntax of Structure

```
struct structure_name  
{ data_type member1;  
  data_type member2;  
  .  
  .  
  data_type member; };
```

Structure Variable Declaration

- When a structure is defined, it creates a user-defined type but, no storage is allocated.
- For the above structure of person, variable can be declared as:

Structure Variable Declaration

```
struct person
{ char name[50];
  int cit_no;
  float salary;
};
```

Inside main function:

```
struct person p1, p2, p[20];
```

```
struct person
{ char name[50];
  int cit_no;
  float salary;
}p1 ,p2 ,p[20];
```

Accessing Members of a Structure

- There are two types of operators used for accessing members of a structure:
 1. Member operator(.)
 2. Structure pointer operator(->) (will be discussed in structure and pointers)

Accessing Members of a Structure -1

- Any member of a structure can be accessed as: `structure_variable_name.member_name`
- Suppose, we want to access salary for variable `p2`. Then, it can be accessed as:

`p2.salary`

Keyword typedef

- Programmer generally use **typedef** while using structure in C language. For example:

```
typedef struct complex
```

```
{ int imag;
```

```
float real; } comp;
```

Inside main:

```
comp c1,c2;
```

- Here, typedef keyword is used in creating a type comp (which is of type as struct complex). Then, two structure variables c1 and c2 are created by this comp type.

Structure and Function

- In C, structure can be passed to functions by two methods:
 1. Passing by value (passing actual value as argument)
 2. Passing by reference (passing address of an argument)

Passing Structure by Value

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

Output

Enter student's name: Kevin Enter
roll number: 149

Output

Name: Kevin
Roll: 149

Enumeration -1

- Is a set of named integer constants that specifies all the legal values that a variable of its type can have.

```
enum color {red, white, blue}
```

```
color C;
```

```
C = red;
```

```
C = white;
```

Enumeration -2

- The key point to understand about an enumeration is that each of the symbols stands for an integer value and can be used in any integer expression.

Enumeration -3

```
#include <stdio.h>
int main()
{
    enum
        Days{Sunday,Monday,Tuesday,Wednesday,Thursday,Friday,Saturday};

    Days TheDay;
    int j = 0;
    printf("Please day of the week (0 to 6)\n");  scanf("%d",&j);
    TheDay = Days(j);

    if(TheDay == Sunday || TheDay == Saturday)
        printf("Hurray it is the weekend\n");
    else
        printf("still at work ");
    return 0;
}
```