MODULE 8: UNION

A Union is a user-defined type similar to structures in C programming, in the sence that it is also a collection of different types of data.

The difference between the two is that UNION stores values of different type in a single location and each members of the structure has its own storage location. In union, space is reserved for the largest data type only. Therefore, union provides a way to manipulate different kinds of data in a single area of storage. If the new assignment is made, the previous value is automatically erased.

The general Syntax for UNION declaration is:

unionkeyword

```
union     union_name
{
   datatype    member1;
   datatype    member2;
   datatype    member3;
      :
};
```

Example

```
union   car
{
 char   name[50];
 int    price;
};
```

Example with union variable

```
union car
{
 char name[50];
 int  price;
} car1, car2, car3;
```

P-9

Difference between unions and structures through an example

```
union   job1
{
  char   name[32];
  float  salary;
  int    workNo;
} ujob;

struct   job2
{
  char   name[32];
  float  salary;
  int    workNO;
} sjob;
```

Size of union = 32

* bcoz largest data type in this union is char name[32], so size of union is 32 bytes.

Size of structure = 32 + 4 + 2 = 38

* in structure, each members has its own storage ie

char array	= 32
float	= 04
int	= 02
total	38 bytes.

Only one union member can be accessed at a time,

you can access all numbers of a structure at once as sufficient memory is allocated for all members. However, its not the case in unions. You can only access a single member of a union at one time.

Example

```
#include <stdio.h>
union job
{
    float salary;
    int   work_no;
} j;
int main()
{
    j.salary = 72000.89;
    j.work_no = 107;
printf("Salary = %f \n", j.salary);
printf("Number of works = %d", j.work_no);
return 0;
}
```