Unions

CHAPTER 34

SURESH TECHS

C PROGRAMMING COURSE

Create a structure to represent IPL team

Team name: RCB
Captain: Depulisis
No of Indian players: 4
Winning percentage: 87

```
struct iplTeam{
    char teamName[40];
    char captainName[20];
    int noOfIndianPlayers;
    float winningPercentage;
};
```

```
union iplTeam{
    char teamName[40];
    char captainName[20];
    int noOfIndianPlayers;
    float winningPercentage;
};
```

68 = 40+20+4+4

40 **Size of maximum member**

Union

- A union can have numerous members, **but only one of them can occupy the memory at any one moment**.
- Unions allow developers to **optimize memory usage** while declaring variables.

```c
#include<stdio.h>
struct Student{
    char name[20];
    int rollNo;
    int marks;
    float percentage;
};

union iplTeam{
    char teamName[40];
    char captainName[20];
    int noOfIndianPlayers;
    float winningPercentage;
};

int main(){
    struct Student student1;
    union iplTeam rcb;
    printf("%d\n",sizeof(student1));
    printf("%d",sizeof(rcb));
    return 0;
}
```

Assigning and accessing values to/from unions

```c
#include<stdio.h>
union iplTeam{
    char teamName[40];
    char captainName[20];
    int noOfIndianPlayers;
    float winningPercentage;
};

int main(){
    union iplTeam rcb;
    strcpy(rcb.teamName,"RCB");
    strcpy(rcb.captainName,"Duplisis");
    rcb.noOfIndianPlayers=4;
    rcb.winningPercentage=87;
    printf("Team: %s\n",rcb.teamName);
    printf("Captain: %s\n",rcb.captainName);
    printf("Number of Indian players: %d\n",rcb.noOfIndianPlayers);
    printf("Winning percentage: %f",rcb.winningPercentage);
    return 0;
}
```

```
Team:
Captain:
Number of Indian players: 1118699520
Winning percentage: 87.000000
```

```c
#include<stdio.h>
union iplTeam{
    char teamName[40];
    char captainName[20];
    int noOfIndianPlayers;
    float winningPercentage;
};

int main(){
    union iplTeam rcb;
    strcpy(rcb.teamName,"RCB");
    printf("Team: %s\n",rcb.teamName);
    strcpy(rcb.captainName,"Duplisis");
    printf("Captain: %s\n",rcb.captainName);
    rcb.noOfIndianPlayers=4;
    printf("Number of Indian players: %d\n",rcb.noOfIndianPlayers);
    rcb.winningPercentage=87;
    printf("Winning percentage: %f",rcb.winningPercentage);
    return 0;
}
```

```
Team: RCB
Captain: Duplisis
Number of Indian players: 4
Winning percentage: 87.000000
```

How is union size calculated?

Minimum size required to hold maximum value of the member

```
union iplTeam{
    char teamName[40];
    char captainName[20];
    int noOfIndianPlayers;
    float winningPercentage;
};
```

40 - Size of maximum member

We can’t initialize multiple members at a time.

```c
#include<stdio.h>
union iplTeam{
    char teamName[40];
    char captainName[20];
    int noOfIndianPlayers;
    float winningPercentage;
};

int main(){
    union iplTeam rcb={"RCB","Duplisis",4,87};
//    strcpy(rcb.teamName,"RCB");
    printf("Team: %s\n",rcb.teamName);
  //  strcpy(rcb.captainName,"Duplisis");
    printf("Captain: %s\n",rcb.captainName);
    //rcb.noOfIndianPlayers=4;
    printf("Number of Indian players: %d\n",rcb.noOfIndianPlayers);
    //rcb.winningPercentage=87;
    printf("Winning percentage: %f",rcb.winningPercentage);
    return 0;
}
```

```
Team: RCB
Captain: RCB
Number of Indian players: 4342610
Winning percentage: 0.000000
```

```c
#include<stdio.h>
union iplTeam{
    char teamName[40];
    char captainName[20];
    int noOfIndianPlayers;
    float winningPercentage;
};

int main(){
    union iplTeam rcb={"RCB"};
//     strcpy(rcb.teamName,"RCB");
    printf("Team: %s\n",rcb.teamName);
    strcpy(rcb.captainName,"Duplisis");
    printf("Captain: %s\n",rcb.captainName);
    rcb.noOfIndianPlayers=4;
    printf("Number of Indian players: %d\n",rcb.noOfIndianPlayers);
    rcb.winningPercentage=87;
    printf("Winning percentage: %f",rcb.winningPercentage);
    return 0;
}
```

```
Team: RCB
Captain: Duplisis
Number of Indian players: 4
Winning percentage: 87.000000
```

Can Initialize only one member at a time

Guess the output

```c
#include<stdio.h>
union iplTeam{
    char teamName[40];
    char captainName[20];
    int noOfIndianPlayers;
    float winningPercentage;
};

int main(){
    union iplTeam rcb={99};
//     strcpy(rcb.teamName,"RCB");
    printf("Team: %s\n",rcb.teamName);
    strcpy(rcb.captainName,"Duplisis");
    printf("Captain: %s\n",rcb.captainName);
    rcb.noOfIndianPlayers=4;
    printf("Number of Indian players: %d\n",rcb.noOfIndianPlayers);
    rcb.winningPercentage=87;
    printf("Winning percentage: %f",rcb.winningPercentage);
    return 0;
}
```

```
Team: c
Captain: Duplisis
Number of Indian players: 4
Winning percentage: 87.000000
```

Note

- Unions are used when memory is very important
- Used in embedded systems
- All the concepts that we discussed in structures applies for union as well
- Ex:
  - nested structures->nested unions
  - Three types of initialization
    - Dot
    - Braces
    - Designated initialization

Difference between structure and union

Structure	Union
struct statement is used to define a structure.	**union** keyword is used to define a union.
Every member is **assigned a unique memory** location.	All the data members **share single memory location.**
Change in the value of one data member does not affect other data members in the structure.	**Change in the value of one data member affects the value of other data members**.
You can **initialize multiple members at a time.**	You can **initialize only the first member at once.**
A structure can store **multiple values of the different members.**	A union stores **one value at a time for all of its members**
A structure's **total size is the sum of the size of every data member.**	A union's **total size is the size of the largest data member.**
Users can **access or retrieve any member at a time.**	You can **access or retrieve only one member at a time.**

What next?

- Pointers