Objective:

- It will help in understanding the use of struct within struct and array of struct.
- Use of global variable.

Task-1

The example, which we discussed today in class.

```
struct Address
{
    char city[30];
    char country[50];
    int streetNo;
    char block[30];
    char colony[100];
};
struct Phone
{
    int intlCode;
    int cityCode;
    int phoneNo;
};
struct Student
{
    char rollNo[15];
    char name[40];
    Address ads;
    Phone ph;
};
```

```
int main()
{
    Student std[10];
    std[0].ph.cityCode = 42;
    std[0].ph.intlCode = 92;
    std[0].ph.phoneNo = 385679;

    //you can also do something like
this
    Address ad;
    strcpy(ad.city,"Lahore");
    strcpy(ad.country,"Pakistan");
    ad.streetNo=23;
    strcpy(ad.block,"West B");
    strcpy(ad.colony,"Mars");

    std[0].ads = ad; //shallow copy safe
for address object

    strcpy(std[0].name,"Ahmed");
    strcpy(std[0].rollNo,"BCSF01M001");
    return 1;
}
```

Task-2

Write a program that stores the following data about a soccer player in a structure:

Player's Name
Player's Number
Points Scored by Player

The program should keep an array of 12 of these structures. Each element is for a different player on a team. When the program runs it should ask the user to enter the data for each player. It should then show a table that lists each player's number, name, and points scored. The program should also calculate and display the total points earned by the team. The number and name of the player who has earned the most points should also be displayed.

Task-3

This application will build software for vending machine through which buyers/clients will be able to draw food items by paying money.

Vending Machine Application

Item struct represents the item details to be stored in machine.

```
struct Item
{
    char name[100];      // name of item
    int noOfItems;       //number of items in machine
    float unitPrice;     // unit price of item
};
```

VendingMachine struct keeps the record of items stored in machine to be sold.

```
const int capacity = 100;
struct VendingMachine
{
    Item data[capacity];      // array of Items in machine: should not have duplicate items:
                              //machine can store only 100 items so array size is fixed i.e 100
    int noOfItems;            // number of unique items stored in machine/array pointed by data
    int rupees;               // currency stored in machine for giving change to user
};
```

***We have used global const identifier 'capacity' in this task, which is not a good approach but soon we shall see better alternatives
***but through this, we shall also get to know global identifier.

→ Functions, which will be used by handler/admin of vending machine

```
void inputItem ( Item & );
void printItem ( Item );
void addNewItemInVendingMachine ( VendingMachine &, Item )
void removeItemFromVendingMachine ( VendingMachine &, char * itemName)
void updateItemUnitPrice (VendingMachine & , char * itemName, int newUnitPrice)
void displayVendingMachineItems ( VendingMachine  )
void addCurrencyInVendingMachine ( VendingMachine &, int rupees);
```

→ Functions, which will be used by buyer/client of items in vending machine

```
void orderItem ( VendingMachine &, int itemNumber );
```

When the machine gets ON, It displays the items on its console in the following manner.

Vending Machine

1 Cadbury……30 Rs.
2 Coffee……40 Rs.
3 Chocolate……10 Rs.
4 Latte……15 Rs.
5 Water Nestle……45 Rs.
6 Cappuccino……50 Rs.
7 Pepsi 50ml…..……40 Rs.

Enter Item Number to Buy: _

Sample Run:
Assume user enter 3, In case of valid number machine asks for rupees to be entered in machine. In return, machine gives remaining balance and asks the user to draw item from machine.

Vending Machine

1 Cadbury……30 Rs.
2 Coffee……40 Rs.
3 Chocolate……10 Rs.
4 Latte……15 Rs.
5 Water Nestle……45 Rs.
6 Cappuccino……50 Rs.
7 Pepsi 50ml……40 Rs.

Enter Item Number to Buy: 3
Enter Rupees: 50 Rs.
Get the remaining balance: 40 Rs.
And Get Item from Machine: Keep Coming ☺

How the handler/Admin of the machine will interact with machine. The admin has a secret code i.e. 101. When he enters this number at the menu "Enter Item Number to Buy", the machine display the interface/menu for the machine admin.

Vending Machine

Press 1 – Add Item in Machine
Press 2 – Remove Item from Machine
Press 3 – Update Item Unit Price
Press 4 – Add Currency in Machine

For Example, if user enters 1 then machine asks for the item details and return to main menu of Admin. If admin enter 101 against any option again then it returns to main menu i.e. displays items in vending machine to take order from user.

Vending Machine

Enter Name of Item: Coca Cola
Enter Count: 20
Enter Unit Price: 50

Item Added Successfully.