# LAB NO. 04

# STRUCTURES IN C++

**Lab Objectives**

Following are the lab objectives:

1. Array of Structures
2. Structure to functions
3. Pointer to structure

## Instructions

- This is individual Lab work/task.
- Complete this lab work within lab timing.
- Discussion with peers is not allowed.
- Copy paste from Internet will give you **negative marks**.
- Lab work is divided into small tasks, complete all tasks sequentially.

# Array of Structure

Structure is collection of different data type. An object of structure represents a single record in memory, if we want more than one record of structure type, we have to create an array of structure or object. As we know, an array is a collection of similar type, therefore an array can be of structure type.

**Syntax for declaring structure array**

```
struct struct-name
{
    datatype var1;
    datatype var2;
    - - - - - - - - - -
    - - - - - - - - - -
    datatype varN;
};
struct-name obj [ size ];
```

**Example for declaring structure array**

```
#include<iostream.h>
struct Employee
{
    int Id;
    char Name[25];
    int Age;
    long Salary;
};
void main()
{
    int i;
    Employee Emp[ 3 ];        //Statement   1
    for(i=0;i<3;i++)
```

```
        {
        cout << "\nEnter details of " << i+1 << " Employee";
            cout << "\n\tEnter Employee Id : ";
            cin >> Emp[i].Id;
            cout << "\n\tEnter Employee Name : ";
            cin >> Emp[i].Name;
            cout << "\n\tEnter Employee Age : ";
            cin >> Emp[i].Age;
            cout << "\n\tEnter Employee Salary : ";
            cin >> Emp[i].Salary;
        }
        cout << "\nDetails of Employees";
        for(i=0;i<3;i++)
        cout << "\n"<< Emp[i].Id <<"\t"<< Emp[i].Name <<"\t"
                << Emp[i].Age <<"\t"<< Emp[i].Salary;
    }
```

## Array within Structure

As we know, structure is collection of different data type. Like normal data type, It can also store an array as well.

Syntax for array within structure

```
struct struct-name
{
    datatype var1;              // normal variable
    datatype array [size];      // array variable
    - - - - - - - - - -
    - - - - - - - - - -
    datatype varN;
};
```

```
struct-name obj;
```

Example for array within structure

```
struct Student
{
    int Roll;
    char Name[25];
    int Marks[3];          //Statement 1 : array of marks
    int Total;
    float Avg;
};
void main()
{
    int i;
    Student S;
    cout << "\n\nEnter Student Roll : ";
    cin >> S.Roll;
    cout << "\n\nEnter Student Name : ";
    cin >> S.Name;
    S.Total = 0;
    for(i=0;i<3;i++)
    {
        cout << "\n\nEnter Marks " << i+1 << " : ";
        cin >> S.Marks[i];
        S.Total = S.Total + S.Marks[i];
    }
    S.Avg = S.Total / 3;

    cout << "\nRoll : " << S.Roll;
```

```
    cout << "\nName : " << S.Name;
    cout << "\nTotal : " << S.Total;
    cout << "\nAverage : " << S.Avg;
}
```

## Structure and Function

Using function we can pass structure as function argument and we can also return structure from function.

### Passing structure as function argument

Structure can be passed to function through its object therefore passing structure to function or passing structure object to function is same thing because structure object represents the structure. Like normal variable, structure variable (structure object) can be pass by value or by references/addresses.

### Passing Structure by Value

In this approach, the structure object is passed as function argument to the definition of function, here object is representing the members of structure with their values.

### Example for passing structure object by value

```
#include<iostream.h>
struct Employee
{
    int Id;
    char Name[25];
    int Age;
    long Salary;
};
void Display(struct Employee);
void main()
{
    Employee Emp = {1,"Kumar",29,45000};
```

```
    Display(Emp);
}
void Display(struct Employee E)
{
        cout << "\n\nEmployee Id : " << E.Id;
        cout << "\nEmployee Name : " << E.Name;
        cout << "\nEmployee Age : " << E.Age;
        cout << "\nEmployee Salary : " << E.Salary;
}
```

Output :

```
    Employee Id : 1
    Employee Name : Kumar
    Employee Age : 29
    Employee Salary : 45000
```

## Passing Structure by Reference

In this approach, the reference/address structure object is passed as function argument to the definition of function.

## Example for passing structure object by reference

```
#include<iostream.h>
struct Employee
{
    int Id;
    char Name[25];
    int Age;
    long Salary;
};
void Display(struct Employee*);
void main()
```

```
{
    Employee Emp = {1,"Kumar",29,45000};
    Display(&Emp);
}

void Display(struct Employee *E)
{
        cout << "\n\nEmployee Id : " << E->Id;
        cout << "\nEmployee Name : " << E->Name;
        cout << "\nEmployee Age : " << E->Age;
        cout << "\nEmployee Salary : " << E->Salary;
}
```

Output :

```
    Employee Id : 1
    Employee Name : Kumar
    Employee Age : 29
    Employee Salary : 45000
```

## Function Returning Structure

Structure is user-defined data type, like built-in data types structure can be return from function.

**Example for passing structure object by reference**

```
 #include<iostream.h>
struct Employee
{
    int Id;
    char Name[25];
    int Age;
    long Salary;
};
```

```
Employee Input();           //Statement   1
void main()
{
    Employee Emp;
    Emp = Input();
        cout << "\n\nEmployee Id : " << E.Id;
        cout << "\nEmployee Name : " << E.Name;
        cout << "\nEmployee Age : " << E.Age;
        cout << "\nEmployee Salary : " << E.Salary;
}
Employee Input()
{
    Employee E;
        cout << "\nEnter Employee Id : ";
        cin >> E.Id;
        cout << "\nEnter Employee Name : ";
        cin >> E.Name;
        cout << "\nEnter Employee Age : ";
        cin >> E.Age;
        cout << "\nEnter Employee Salary : ";
        cin >> E.Salary;
    return E;           //Statement   2
}
```

Output :

```
    Enter Employee Id : 10
    Enter Employee Name : Ajay
    Enter Employee Age : 25
    Enter Employee Salary : 15000
```

Employee Id : 10

Employee Name : Ajay

Employee Age : 25

Employee Salary : 15000

## Pointers to Structure

It's possible to create a pointer that points to a structure. It is similar to how pointers pointing to native data types like int, float, double, etc. are created. Note that a pointer in C++ will store a memory location.

### Example:

```
#include <iostream>
using namespace std;
struct Length
{
	int meters;
	float centimeters;
};
int main()
{
	Length *ptr, l;
	ptr = &l;
	cout << "Enter meters: ";
	cin >> (*ptr).meters;
	cout << "Enter centimeters: ";
	cin >> (*ptr).centimeters;
	cout << "Length = " << (*ptr).meters << " meters " << (*ptr).centimeters << "
centimeters";

	return 0;
}
```

# Lab Tasks

**Q1. Write a structure to store the roll no, name and age (between 10 to 15 ) of students (more than 5). Store the information of the students.**

**1 - Write a function to print the names of all the students having age 14.**

**2 - Write another function to print the names of all the students having even roll no.**

**3 - Write another function to display the details of the student whose roll no is given (i.e. roll no. entered by the user).**

**Q2. Write a structure to store the name, account number and balance of customers (more than 10) and store their information.**

**1 - Write a function to print the names of all the customers having balance less than 20k.**
**2 - Write a function to add 1000k in the balance of all the customers having more than 30k in their balance and then print the incremented value of their balance.**

# Home Activity

**Q3. Let us work on the menu of a library. Create a structure containing book information like accession number, name of author, book title and flag to know whether book is issued or not.**

**Create a menu in which the following can be done.**

**1 - Display book information**
**2 - Add a new book**
**3 - Display all the books in the library of a particular author**
**4 - Display the number of books of a particular title**
**5 - Display the total number of books in the library**
**6 - Issue a book**
**(If we issue a book, then its number gets decreased by 1 and if we add a book, its number gets increased by 1)**

**Q4. Write a structure to store the names, salary and hours of work per day of 10 employees in a company. Write a program to increase the salary depending on the number of hours of work per day as follows and then print the name of all the employees along with their final salaries.**

| | | | |
|---|---|---|---|
| **Hours of work per day** | **8** | **10** | **>=12** |
| **Increase in salary** | **1k** | **2k** | **3k** |