

LAB 16: Structures

Objective(s):

To Understand about:

1. Apply Structures in C++
2. Use of Structures along with previously used concepts of C++.

CLO: CLO1, CLO4

Introduction

Structure is a collection of variables of different data types under a single name. It is similar to a class in that, both holds a collection of data of different data types.

For example: You want to store some information about a person: his/her name, citizenship number and salary. You can easily create different variables name, citNo, salary to store these information's separately. However, in the future, you would want to store information about multiple persons. Now, you'd need to create different variables for each information per person: name1, citNo1, salary1, name2, citNo2, salary2. You can easily visualize how big and messy the code would look. Also, since no relation between the variables (information) would exist, it's going to be a daunting task.

A better approach will be to have a collection of all related information under a single name Person, and use it for every person. Now, the code looks much cleaner, readable and efficient as well. This collection of all related information under a single name Person is a structure.

Declaration of a Structure in C++

Then inside the curly braces, you can declare one or more members (declare variables inside curly braces) of that structure. For example:

```
struct Person
{
    char name[50];
    int age;
    float salary;
};
```

Here a structure person is defined which has three members: name, age and salary.

When a structure is created, no memory is allocated. The structure definition is only the blueprint for the creating of variables. You can imagine it as a datatype. When you define an integer as below:

```
int foo;
```

The int specifies that, variable foo can hold integer element only. Similarly, structure definition only specifies that, what property a structure variable holds when it is defined.

Defining Variable of a Structure

Once you declare a structure person as above. You can define a structure variable as:

```
Person bill;
```

Here, a structure variable bill is defined which is of type structure Person.

When structure variable is defined, only then the required memory is allocated by the compiler.

Considering you have either 32-bit or 64-bit system, the memory of float is 4 bytes, memory of int is 4 bytes and memory of char is 1 byte.

Hence, 58 bytes of memory is allocated for structure variable bill.

Accessing members of a Structure

The members of structure variable is accessed using a dot (.) operator.

Suppose, you want to access age of structure variable bill and assign it 50 to it. You can perform this task by using following code below:

```
bill.age = 50;
```

Example:

```
#include <iostream>
using namespace std;

struct Person
{
    char name[50];
    int age;
    float salary;
};

int main()
{
    Person p1;

    cout << "Enter Full name: ";
    cin.get(p1.name, 50);
    cout << "Enter age: ";
    cin >> p1.age;
    cout << "Enter salary: ";
    cin >> p1.salary;

    cout << "\nDisplaying Information." << endl;
    cout << "Name: " << p1.name << endl;
```

```

    cout << "Age: " << p1.age << endl;
    cout << "Salary: " << p1.salary;

    return 0;
}

```

Output

```

Enter Full name: Magdalena Dankova
Enter age: 27
Enter salary: 1024.4

Displaying Information.
Name: Magdalena Dankova
Age: 27
Salary: 1024.4

```

Structure and Functions

Structure variables can be passed to a function and returned in a similar way as normal arguments.

Example:

```

#include <iostream>
using namespace std;

struct Person
{
    char name[50];
    int age;
    float salary;
};

void displayData(Person); // Function declaration

int main()
{
    Person p;

    cout << "Enter Full name: ";
    cin.get(p.name, 50);
    cout << "Enter age: ";
    cin >> p.age;
    cout << "Enter salary: ";
    cin >> p.salary;

    // Function call with structure variable as an argument
}

```

```

        displayData(p);

        return 0;
    }

    void displayData(Person p)
    {
        cout << "\nDisplaying Information." << endl;
        cout << "Name: " << p.name << endl;
        cout << "Age: " << p.age << endl;
        cout << "Salary: " << p.salary;
    }
}

```

Output

```

Enter Full name: Bill Jobs
Enter age: 55
Enter salary: 34233.4

Displaying Information.
Name: Bill Jobs
Age: 55
Salary: 34233.4

```

In this program, user is asked to enter the name, age and salary of a Person inside main() function. Then, the structure variable p is to be passed to a function using.

```
displayData(p);
```

The return type of displayData() is void and a single argument of type structure Person is passed. Then the members of structure p are displayed from this function.

Lab Tasks:

Task 1

Write a C++ program that maintains record of students. Student contains following details:

- ID
- Name
- Department
- Email
- Phone no

Create a structure of student. Ask user to enter record for 5 students. Store those details in variables of student's type. Print those records on screen.

Task 2

Write a C++ program to do the following:

- Set up a structure/record to store products; each product has a name, a model number and a price.
- Choose appropriate types for these fields.

- Your program should contain a loop which allows entry via the keyboard of up to 10 products, to be stored in an array.
- The loop can be stopped by entry of "quit" for the product name
- Your program should include a function to display all the products details after the entry loop has finished.

Task 3

Write a C++ program that compute Net Salary of Employee. Program contains two user defined functions *empSalary()* and *display()*.

- Create a structure of Employee that contains following data members: EmployeeNumber, Name, BasicSalary, HouseAllowance, MedicalAllowance, Tax, GrossPay and NetSalary
- Employee number, name and basic salary must be taken input from the user.
- *empSalary()* compute salary with given criteria
 - House Allowance = 10% of Basic Salary
 - Medical Allowance = 5% of Basic Salary
 - Tax = 4 % of Basic Salary
 - GrossSalary = Basic+HouseAllowance+MedicalAllowance
 - NetSalary = GrossSalary – Tax
- *display()* for displaying details of Employee

```
#Sample Output
Enter the Employee Number :10129
Enter the employee name: Ahmed Ali
Enter the Basic Salary: 16500

*****
      EMPLOYERS SALARY DETAILS
*****
Employee Number:10129
Employee Name: Ahmed Ali
Basic Salary: 16500
House Allowance:1650
Medical Allowance: 825
Gross Salary: 18975
Tax Deduction: 660
Net Salary: 18315
```