

Week #4

Objective:

The objective of this lab is to introduce students to the concept of structures in C++ and familiarize them with their usage. By the end of this lab, students should be able to understand how to define, initialize, and manipulate structures, as well as use them effectively in your programs.

Introduction to Structures:

In C++, a structure is a user-defined data type that allows you to group together variables of different data types under a single name. Structures are used to represent a collection of related data items, making it easier to manage and manipulate complex data structures in your programs.

Key Concepts:

Definition of Structures: Structures are defined using the **struct** keyword followed by the structure name and a set of members enclosed in curly braces.

```
struct StructureName {  
    dataType member1;  
    dataType member2;  
    // More members...  
};
```

```
struct Person {  
    string name;  
    int age;  
    float height;  
};
```

Structure Members: Each variable inside a structure is called a member or field, and they can have different data types.

Accessing Structure Members: Members of a structure can be accessed using the dot (.) operator.

```
Person p1;  
p1.name = "John";  
p1.age = 30;  
p1.height = 6.1;
```

Nested Structures: Structures can be nested within other structures, allowing for the representation of hierarchical data.

```
struct Address {  
    string street;  
    string city;  
};  
  
struct Employee {  
    string name;  
    int age;  
    Address address;  
};
```

Arrays of Structures: Arrays of structures enable the storage of multiple instances of the same type of data. You can create arrays of structures to store multiple instances of the same type of data. Each element of the array is a structure variable. Arrays of structures are useful for managing collections of related data.

Pointers to Structures: Pointers can be used to dynamically access and manipulate structures. Pointers can be used to access structures dynamically and efficiently. Pointer notation (->) is used to access members of a structure through a pointer.

Use Cases of Structures:

Structures are commonly used to represent real-world entities such as employees, students, books, etc. They are useful for organizing and managing data in a structured manner. Structures facilitate code readability, maintainability, and modularity.

By practicing these concepts through hands-on exercises and challenges, the students will enhance their problem-solving skills and develop a deeper understanding of how to write efficient and effective C++ code. So, let's get started and embark on this journey to master the art of programming logic!

Exercise:

Q.1) Define a structure named Person with members name and age. Create a variable of this structure type, initialize it, and display its contents.

CODE:

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

int startlab3(){
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Start of Lab 03" << endl;
    return 0;
}

struct Person{
    string name;
    int age;
};

int l3q1(){
    Person p1;
    p1.name = "Saad Ali Khan";
    p1.age = 18;
    cout << "Name: " << p1.name << endl;
    cout << "Age: " << p1.age << endl;
    return 0;
}

int main(){
    startlab3();
    l3q1();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Start of Lab 03
Name: Saad Ali Khan
Age: 18
PS D:\SE\oops_labs> █
```

Q.2) Define a structure named Rectangle with members length and width. Create a variable of this structure type, input values for length and width, calculate the area, and display it.

CODE:

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

int startlab3()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Start of Lab 03" << endl;
    return 0;
}

struct Rectangle
{
    float length;
    float width;
};

int l3q2()
{
    Rectangle r1;
    cout << "Enter length: ";
    cin >> r1.length;
    cout << "Enter width: ";
    cin >> r1.width;
    cout << "Area: " << r1.width * r1.length << endl;

    return 0;
}

int main()
{
    startlab3();
    l3q2();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Start of Lab 03
Enter length: 4.5
Enter width: 5.6
Area: 25.2
PS D:\SE\oops_labs> █
```

Q3) Define a structure named Number with a single member num. Create two variables of this structure type, input values for them, swap their values using a function, and display the swapped values

CODE:

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

int startlab3()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Start of Lab 03" << endl;
    return 0;
}

struct Number
{
    int num;
};

int l3q3()
{
    Number num1, num2;
    cout << "Enter number 1: ";
    cin >> num1.num;
    cout << "Enter number 2: ";
    cin >> num2.num;
    cout << "Before Swap: " << endl;
    cout << "num1 = " << num1.num << endl;
    cout << "num2 = " << num2.num << endl;
    swap(num1, num2);
    cout << "After Swap: " << endl;
    cout << "num1 = " << num1.num << endl;
    cout << "num2 = " << num2.num << endl;
}
```

```
        return 0;
    }
int swap(Number n1, Number n2)
{
    Number temp;
    temp = n1;
    n1 = n2;
    n2 = temp;
    cout << "Before Swap: " << endl;
    cout << "num1 = " << n1.num << endl;
    cout << "num2 = " << n2.num << endl;
    return 0;
}
int main()
{
    startlab3();
    l3q3();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Start of Lab 03
Enter number 1: 4
Enter number 2: 5
Before Swap:
num1 = 4
num2 = 5
After Swap:
num1 = 5
num2 = 4
PS D:\SE\oops_labs>
```

Q4 Define a structure named Student with members rollNumber, name, age, and marks. Create an array of 3 Student structures, input data for each student, display their details, and find the student with the highest marks.

CODE:

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

int startlab3()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Start of Lab 03" << endl;
    return 0;
}

struct Student
{
    int rollNumber;
    string name;
    int age;
    int marks;
};

int l3q4()
{
    Student s1[4] = {83, "Saad Ali Khan", 18, 97};
    Student s2[4] = {66, "Muzammil Ali Khan", 22, 94};
    Student s3[4] = {44, "Anus Ali Khan", 24, 96};
    cout << "Roll No: " << s1->rollNumber << "\t"
         << "Name: " << s1->name << "\t"
         << "Age:" << s1->age << "\t"
         << "Marks: " << s1->marks << endl;
    cout << "Roll No: " << s2->rollNumber << "\t"
         << "Name: " << s2->name << "\t"
         << "Age:" << s2->age << "\t"
         << "Marks: " << s2->marks << endl;
    cout << "Roll No: " << s3->rollNumber << "\t"
         << "Name: " << s3->name << "\t"
         << "Age:" << s3->age << "\t"
         << "Marks: " << s3->marks << endl;
```

```
        if (s1->marks > s2->marks && s1->marks > s3->marks)
        {
            cout << "Saad Ali Khan has highest marks which are:" <<
s1->marks;
        }
        else if (s2->marks > s1->marks && s2->marks > s3->marks)
        {
            cout << "Muzammil Ali Khan has highest marks which are:" <<
s2->marks;
        }
        else
        {
            cout << "Anus Ali Khan has highest marks which are:" <<
s3->marks;
        }
        return 0;
    }

int main()
{
    startlab3();
    l3q4();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Start of Lab 03
Roll No: 83      Name: Saad Ali Khan      Age:18  Marks: 97
Roll No: 66      Name: Muzammil Ali Khan Age:22  Marks: 94
Roll No: 44      Name: Anus Ali Khan      Age:24  Marks: 96
Saad Ali Khan has highest marks which are:97
PS D:\SE\oops_labs>
```


Q5) Define a structure named Employee with members employeeID, name, department, and salary. Create an array of 5 Employee structures, input data for each employee, display their details, and calculate the average salary of all employees.

CODE:

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

int startlab3()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Start of Lab 03" << endl;
    return 0;
}

struct Employee
{
    int employeeID;
    string name;
    string department;
    float salary;
};

int l3q5()
{
    Employee e1[4] = {83, "Saad Ali Khan", "HR", 97000};
    Employee e2[4] = {66, "Muzammil Ali Khan", "Testing", 94000};
    Employee e3[4] = {44, "Anus Ali Khan", "Production", 96000};
    Employee e4[4] = {54, "Maaz Ali Murtaza", "QA", 98000};
    Employee e5[4] = {95, "Sakhawat Ali Khan", "Production", 91000};
    cout << "Employee ID: " << e1->employeeID << "\t"
         << "Name: " << e1->name << "\t"
         << "Department:" << e1->department << "\t\t"
         << "Salary: " << e1->salary << endl;
    cout << "Employee ID: " << e2->employeeID << "\t"
         << "Name: " << e2->name << "\t"
         << "Department:" << e2->department << "\t"
         << "Salary: " << e2->salary << endl;
    cout << "Employee ID: " << e3->employeeID << "\t"
         << "Name: " << e3->name << "\t"
         << "Department:" << e3->department << "\t"
         << "Salary: " << e3->salary << endl;
```

```
    cout << "Employee ID: " << e4->employeeID << "\t"
        << "Name: " << e4->name << "\t"
        << "Department:" << e4->department << "\t\t"
        << "Salary: " << e4->salary << endl;
    cout << "Employee ID: " << e5->employeeID << "\t"
        << "Name: " << e5->name << "\t"
        << "Department:" << e5->department << "\t"
        << "Salary: " << e5->salary << endl;

    float average_salary = (e1->salary + e2->salary + e3->salary +
e4->salary + e5->salary) / 5;
    cout << "The average salary of all employees is: " <<
average_salary;

    return 0;
}

int main()
{
    startlab3();
    l3q5();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Start of Lab 03
Employee ID: 83 Name: Saad Ali Khan    Department:HR           Salary: 97000
Employee ID: 66 Name: Muzammil Ali Khan Department:Testing      Salary: 94000
Employee ID: 44 Name: Anus Ali Khan    Department:Production    Salary: 96000
Employee ID: 54 Name: Maaz Ali Murtaza  Department:QA            Salary: 98000
Employee ID: 95 Name: Sakhawat Ali Khan Department:Production    Salary: 91000
The average salary of all employees is: 95200
PS D:\SE\oops_labs>
```

Q6) Define a structure named Book with members title, author, genre, and yearOfPublication. Create an array of 3 Book structures, input data for each book, display their details, and find the book with the earliest publication year.

CODE:

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

int startlab3()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Start of Lab 03" << endl;
    return 0;
}

struct Book
{
    string title;
    string author;
    string genre;
    int year_of_publictaion;
};

int l3q6()
{
    Book b1[4] = {"Humanity", "Saad Ali Khan", "Drama", 2024};
    Book b2[4] = {"Art of War", "Muzammil Ali Khan", "Drama,War",
2000};
    Book b3[4] = {"Chemical of life", "Anus Ali Khan", "Sci-fi",
2002};

    cout << "Title: " << b1->title << "\t\t"
        << "Author: " << b1->author << "\t\t"
        << "Genre:" << b1->genre << "\t"
        << "Publication: " << b1->year_of_publictaion << endl;
    cout << "Title: " << b2->title << "\t"
        << "Author: " << b2->author << "\t"
        << "Genre:" << b2->genre << "\t"
        << "Publication: " << b2->year_of_publictaion << endl;
    cout << "Title: " << b3->title << "\t"
        << "Author: " << b3->author << "\t\t"
        << "Genre:" << b3->genre << "\t"
```

```
<< "Publication: " << b3->year_of_publication << endl;
    if (b1->year_of_publication < b2->year_of_publication &&
b1->year_of_publication < b3->year_of_publication)
    {
        cout << "Book 1 was released before book 2 and book 3 in
year:" << b1->year_of_publication;
    }
    else if (b2->year_of_publication < b1->year_of_publication &&
b2->year_of_publication < b3->year_of_publication)
    {
        cout << "Book 2 was released before book 1 and book 3 in
year:" << b2->year_of_publication;
    }
    else
    {
        cout << "Book 3 was released before book 1 and book 2 in
year:" << b3->year_of_publication;
    }
    return 0;
}

int main()
{
    startlab3();
    l3q6();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Start of Lab 03
Title: Humanity          Author: Saad Ali Khan          Genre:Drama      Publication: 2024
Title: Art of War        Author: Muzammil Ali Khan        Genre:Drama,War  Publication: 2000
Title: Chemical of life  Author: Anus Ali Khan          Genre:Sci-fi     Publication: 2002
Book 2 was released before book 1 and book 3 in year:2000
PS D:\SE\oops_labs> |
```

Q7) Define a structure named Product with members productName, price, and quantity. Create a variable of this structure type, input data for the product, calculate the total cost, and display it.

CODE:

```
#include <iostream>
using namespace std;
int startlab3()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Start of Lab 03" << endl;
    return 0;
}
struct Product
{
    string productName;
    float price;
    int quantity;
};

int l3q7()
{
    Product p1;
    p1.productName = "Milk";
    p1.price = 200;
    p1.quantity = 3;
    cout << "The total cost of " << p1.productName << " having
quantity " << p1.quantity << " is: " << p1.price * p1.quantity;
    return 0;
}
int main(){
    startlab3();
    l3q7();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Start of Lab 03
The total cost of Milk having quantity 3 is: 600
PS D:\SE\oops_labs> █
```