Class Vs. Structure:

In C++, both **class** and **structure** (struct) are user-defined data types used to group variables and functions. However, they have some key differences:

Key Differences Between Class and Structure in C++

Feature	Class	Structure
Access Specifier (Default)	Private	Public
Encapsulation	Supports data hiding (OOP principle)	Does not support data hiding by default
Inheritance	Supports inheritance	Supports inheritance but defaults to **public** instead of **private**
Usage	Typically used for complex data and OOP	Used for simple data grouping
Member Functions	Can have member functions	Can have member functions
Memory Allocation	No difference	No difference

Example: Class vs. Structure

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

// Structure Example
struct Student {
    string name;
    int age;

    // Function inside struct
    void display() {
        cout << "Student Name: " << name << ", Age: " << age << endl;
    }
};

// Class Example
class Employee {
private:
    string name;
    int salary;
```

```
public:
    // Constructor
    Employee(string empName, int empSalary) {
        name = empName;
        salary = empSalary;
    }

    // Member function
    void display() {
        cout << "Employee Name: " << name << ", Salary: " << salary << endl;
    }
};

int main() {
    // Structure usage
    Student s1;
    s1.name = "Alice";  // Allowed (public by default)
    s1.age = 20;
    s1.display();

    // Class usage
    Employee e1("Bob", 50000);
    // e1.name = "Bob"; // Not allowed (private by default)
    e1.display();

    return 0;
}
```

```
Student Name: Alice, Age: 20
Employee Name: Bob, Salary: 50000
```

- Use **struct** for simple data structures where data members are public.
- Use **class** for **Object-Oriented Programming (OOP)** with encapsulation and data hiding.