

# Assignment - 3

## C++

**Name:** Kamithkar Vinod

**Course:** PG DAC AUGUST 2025

**PRN:** 250850320040

**Form No:** 250500480

**Date:** 09-10-2025

---

### Problem 1: Basic Class

**Task:** Write a program in C++ to create a class Car with data members name and speed.

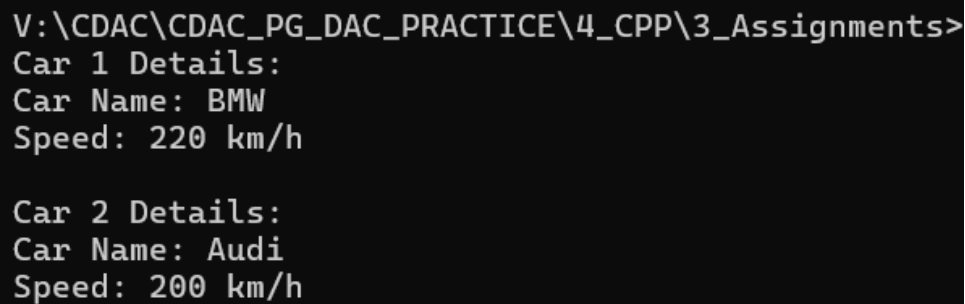
- Use a member function display() to print values.
- Create two objects and display their details.

**Code:** —

```
1 #include <iostream>
2 using namespace std;
3
4 class Car {
5     string name;
6     int speed;
7
8 public:
9     void setData(string n, int s) {
10         name = n;
11         speed = s;
12     }
13
14     void display() {
15         cout << "Car Name: " << name << endl;
16         cout << "Speed: " << speed << " km/h" << endl;
17     }
18 };
19
20 int main() {
21     Car c1, c2;
22     c1.setData("BMW", 220);
23     c2.setData("Audi", 200);
```

```
24
25     cout << "Car 1 Details:" << endl;
26     c1.display();
27     cout << "\nCar 2 Details:" << endl;
28     c2.display();
29     return 0;
30 }
```

Output: 1—



```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\3_Assignments>
Car 1 Details:
Car Name: BMW
Speed: 220 km/h

Car 2 Details:
Car Name: Audi
Speed: 200 km/h
```

## Problem 2: Rectangle (Area Perimeter)

**Task:** Create a class Rectangle with data members length and width.

Add member functions to calculate area and perimeter.

Read values from user and display results.

Code: —

```
1 #include <iostream>
2 using namespace std;
3
4 class Rectangle {
5     float length, width;
6
7 public:
8     void input() {
9         cout << "Enter length and width: ";
10        cin >> length >> width;
11    }
12
13    float area() {
14        return length * width;
15    }
16
17    float perimeter() {
18        return 2 * (length + width);
19    }
20
21    void display() {
22        cout << "Area: " << area() << endl;
```

```
23         cout << "Perimeter: " << perimeter() << endl;
24     }
25 };
26
27 int main() {
28     Rectangle r;
29     r.input();
30     r.display();
31     return 0;
32 }
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\3_Assignments>
Enter length and width: 4 5
Area: 20
Perimeter: 18
```

### Problem 3: Student Details

**Task:** Create a class Student with data members rollNo, name, and marks.

Add member function input() to take values.

Add function display() to print them.

Create an array of 3 students and display all details.

Code: —

```
1 #include <iostream>
2 using namespace std;
3
4 class Student {
5     int rollNo;
6     string name;
7     float marks;
8
9 public:
10     void input() {
11         cout << "Enter Roll No, Name, and Marks: ";
12         cin >> rollNo >> name >> marks;
13     }
14
15     void display() {
16         cout << "Roll No: " << rollNo << ", Name: " << name << ",
17             Marks: " << marks << endl;
18     }
19 };
20
21 int main() {
```

```
21     Student s[3];
22     cout << "Enter details for 3 students:\n";
23     for (int i = 0; i < 3; i++) {
24         s[i].input();
25     }
26
27     cout << "\nStudent Details:\n";
28     for (int i = 0; i < 3; i++) {
29         s[i].display();
30     }
31     return 0;
32 }
```

**Output:** —

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\3_Assignments>
Enter details for 3 students:
Enter Roll No, Name, and Marks: 101 Vinod 99
Enter Roll No, Name, and Marks: 102 Sony 98
Enter Roll No, Name, and Marks: 103 Navvu 97

Student Details:
Roll No: 101, Name: Vinod, Marks: 99
Roll No: 102, Name: Sony, Marks: 98
Roll No: 103, Name: Navvu, Marks: 97
```

## Problem 4: Bank Account

**Task:** Create a class BankAccount with:

Data members: accountNumber, balance.

Functions: deposit(), withdraw(), displayBalance().

Perform deposit and withdrawal operations using objects.

**Code:** —

```
1 #include <iostream>
2 using namespace std;
3
4 class BankAccount {
5     int accountNumber;
6     double balance;
7
8 public:
9     void openAccount(int accNo, double bal) {
10         accountNumber = accNo;
11         balance = bal;
12     }
13 }
```

```
14 void deposit(double amount) {
15     balance += amount;
16     cout << "Deposited: " << amount << endl;
17 }
18
19 void withdraw(double amount) {
20     if (amount <= balance) {
21         balance -= amount;
22         cout << "Withdrawn: " << amount << endl;
23     } else {
24         cout << "Insufficient balance!" << endl;
25     }
26 }
27
28 void displayBalance() {
29     cout << "Account Number: " << accountNumber << ", Balance
30         : " << balance << endl;
31 }
32
33 int main() {
34     BankAccount acc1;
35     acc1.openAccount(1001, 5000);
36     acc1.deposit(2000);
37     acc1.withdraw(1500);
38     acc1.displayBalance();
39     return 0;
40 }
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\3_Assignments>
Deposited: 2000
Withdrawn: 1500
Account Number: 1001, Balance: 5500
```

### Problem 5: Employee Salary (Parameterized Constructor)

**Task:** Write a C++ program to create a class Employee with data members id, name, and salary.

Use a parameterized constructor to initialize values.

Display employee details using a function.

Code: —

```
1 #include <iostream>
2 using namespace std;
3
4 class Employee {
```

```
5     int id;
6     string name;
7     float salary;
8
9 public:
10    Employee(int i, string n, float s) {
11        id = i;
12        name = n;
13        salary = s;
14    }
15
16    void display() {
17        cout << "ID: " << id << ", Name: " << name << ", Salary:
18            " << salary << endl;
19    }
20 };
21
22 int main() {
23     Employee e1(101, "Vinod", 50000);
24     Employee e2(102, "Kumar", 60000);
25
26     e1.display();
27     e2.display();
28     return 0;
29 }
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\3_Assignments>
ID: 101, Name: Vinod, Salary: 50000
ID: 102, Name: Kumar, Salary: 60000
```

## Problem 6: Complex Number (Object as Argument)

**Task:** Create a class Complex with data members real and imag.

Add a member function add() that takes another Complex object and returns the result as a new object.

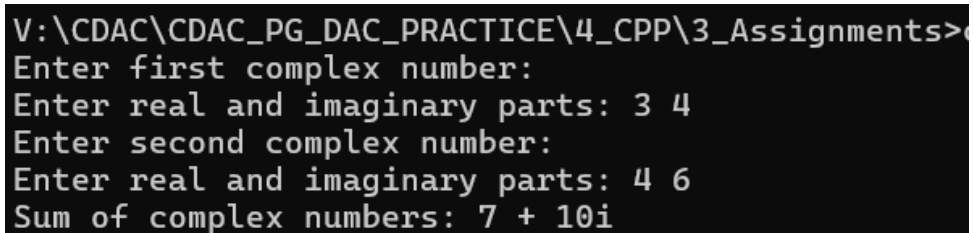
Display the sum of two complex numbers.

Code: —

```
1 #include <iostream>
2 using namespace std;
3
4 class Complex {
5     float real, imag;
6
7 public:
8     void input() {
```

```
9      cout << "Enter real and imaginary parts: ";
10     cin >> real >> imag;
11 }
12
13 Complex add(Complex c2) {
14     Complex temp;
15     temp.real = real + c2.real;
16     temp.imag = imag + c2.imag;
17     return temp;
18 }
19
20 void display() {
21     cout << real << " + " << imag << "i" << endl;
22 }
23 };
24
25 int main() {
26     Complex c1, c2, c3;
27     cout << "Enter first complex number:\n";
28     c1.input();
29     cout << "Enter second complex number:\n";
30     c2.input();
31
32     c3 = c1.add(c2);
33
34     cout << "Sum of complex numbers: ";
35     c3.display();
36     return 0;
37 }
```

Output: —



```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\3_Assignments>
Enter first complex number:
Enter real and imaginary parts: 3 4
Enter second complex number:
Enter real and imaginary parts: 4 6
Sum of complex numbers: 7 + 10i
```

## Problem 7: Library Management

**Task:** Create a class Book with data members title, author, and price.

Write functions to input and display details.

Create an array of 5 books and print the book with the highest price.

Code: —

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

```
3
4 class Book {
5     string title, author;
6     float price;
7
8 public:
9     void input() {
10         cout << "Enter Title, Author, and Price: ";
11         cin >> title >> author >> price;
12     }
13
14     void display() {
15         cout << "Title: " << title << ", Author: " << author <<
16             ", Price: " << price << endl;
17     }
18
19     float getPrice() {
20         return price;
21     }
22 };
23
24 int main() {
25     Book b[5];
26     cout << "Enter details for 5 books:\n";
27     for (int i = 0; i < 5; i++) {
28         b[i].input();
29     }
30
31     int maxIndex = 0;
32     for (int i = 1; i < 5; i++) {
33         if (b[i].getPrice() > b[maxIndex].getPrice())
34             maxIndex = i;
35     }
36
37     cout << "\nBook with Highest Price:\n";
38     b[maxIndex].display();
39     return 0;
40 }
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\3_Assignments>libraryManagement
Enter details for 5 books:
Enter Title, Author, and Price: c DennisRithchie 750
Enter Title, Author, and Price: c++ BjarneStroustop 850
Enter Title, Author, and Price: java jamesGosling 950
Enter Title, Author, and Price: python guidoVanRussum 1050
Enter Title, Author, and Price: c# andresHejlesberg 1150

Book with Highest Price:
Title: c#, Author: andresHejlesberg, Price: 1150
```