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:

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

class Car {
    string name;
    int speed;

public:
    void setData(string n, int s) {
        name = n;
        speed = s;
    }

    void display() {
        cout << "Car Name: " << name << endl;
        cout << "Speed: " << speed << " km/h" << endl;
    }
};

int main() {
    Car c1, c2;
    c1.setData("BMW", 220);
    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;
```

```
        cout << "Perimeter: " << perimeter() << endl;
    }
};

int main() {
    Rectangle r;
    r.input();
    r.display();
    return 0;
}
```

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: —

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

class Student {
    int rollNo;
    string name;
    float marks;

public:
    void input() {
        cout << "Enter Roll No, Name, and Marks: ";
        cin >> rollNo >> name >> marks;
    }

    void display() {
        cout << "Roll No: " << rollNo << ", Name: " << name << ",
            Marks: " << marks << endl;
    }
};

int main() {
```

```
    Student s[3];
    cout << "Enter details for 3 students:\n";
    for (int i = 0; i < 3; i++) {
        s[i].input();
    }

    cout << "\nStudent Details:\n";
    for (int i = 0; i < 3; i++) {
        s[i].display();
    }
    return 0;
}
```

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: —

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

class BankAccount {
    int accountNumber;
    double balance;

public:
    void openAccount(int accNo, double bal) {
        accountNumber = accNo;
        balance = bal;
    }

```

```
    void deposit(double amount) {
        balance += amount;
        cout << "Deposited: " << amount << endl;
    }

    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            cout << "Withdrawn: " << amount << endl;
        } else {
            cout << "Insufficient balance!" << endl;
        }
    }

    void displayBalance() {
        cout << "Account Number: " << accountNumber << ", Balance
           : " << balance << endl;
    }
};

int main() {
    BankAccount acc1;
    acc1.openAccount(1001, 5000);
    acc1.deposit(2000);
    acc1.withdraw(1500);
    acc1.displayBalance();
    return 0;
}
```

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: —

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

class Employee {
```

```
    int id;
    string name;
    float salary;

public:
    Employee(int i, string n, float s) {
        id = i;
        name = n;
        salary = s;
    }

    void display() {
        cout << "ID: " << id << ", Name: " << name << ", Salary:
           " << salary << endl;
    }
};

int main() {
    Employee e1(101, "Vinod", 50000);
    Employee e2(102, "Kumar", 60000);

    e1.display();
    e2.display();
    return 0;
}
```

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: —

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

class Complex {
    float real, imag;

public:
    void input() {
```

```
        cout << "Enter real and imaginary parts: ";
        cin >> real >> imag;
    }

    Complex add(Complex c2) {
        Complex temp;
        temp.real = real + c2.real;
        temp.imag = imag + c2.imag;
        return temp;
    }

    void display() {
        cout << real << " + " << imag << "i" << endl;
    }
};

int main() {
    Complex c1, c2, c3;
    cout << "Enter first complex number:\n";
    c1.input();
    cout << "Enter second complex number:\n";
    c2.input();

    c3 = c1.add(c2);

    cout << "Sum of complex numbers: ";
    c3.display();
    return 0;
}
```

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: —

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

```cpp

class Book {
    string title, author;
    float price;

public:
    void input() {
        cout << "Enter Title, Author, and Price: ";
        cin >> title >> author >> price;
    }

    void display() {
        cout << "Title: " << title << ", Author: " << author <<
           ", Price: " << price << endl;
    }

    float getPrice() {
        return price;
    }
};

int main() {
    Book b[5];
    cout << "Enter details for 5 books:\n";
    for (int i = 0; i < 5; i++) {
        b[i].input();
    }

    int maxIndex = 0;
    for (int i = 1; i < 5; i++) {
        if (b[i].getPrice() > b[maxIndex].getPrice())
            maxIndex = i;
    }

    cout << "\nBook with Highest Price:\n";
    b[maxIndex].display();
    return 0;
}
```

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
```