# CPP-A17: Encapsulation, and Inheritance

## Instructions for Students:

👉 Create **fully encapsulated classes** for each problem.

👉 Use **private data members** and **public getter/setter functions** to access or modify the data.

👉 Think where you can use **inheritance** (Hint provided where necessary).

👉 Write neat, clean, and modular code.

👉 Provide proper output format as shown in the expected output.

## 🔥 Problem 1: Library Book Record System

**Problem Statement:**

Create a system that stores book details like Book ID, Title, Author Name, and Price. The system should display the book details.

**Sample Input:**

```
Enter Book ID: 101
Enter Book Title: C++ Programming
Enter Author Name: Bjarne Stroustrup
Enter Price: 500
```

**Expected Output:**

```
Book Details:
ID: 101
Title: C++ Programming
Author: Bjarne Stroustrup
Price: 500
```

# 🔥 Problem 2: Car Showroom Inventory

**Problem Statement:**

Create a system to store and display car details like Brand, Model, and Price.

**Sample Input:**

```
Enter Brand: Toyota
Enter Model: Fortuner
Enter Price: 3500000
```

**Expected Output:**

```
Car Details:
Brand: Toyota
Model: Fortuner
Price: 3500000
```

# 🔥 Problem 3: Student Sports Record

**Problem Statement:**

Create a system to store a student's personal and sports performance details like ID, Name, Sport, and Rank.

**Sample Input:**

```
Enter Student ID: 23
Enter Name: Raj
Enter Sport: Badminton
Enter Rank: 2
```

**Expected Output:**

```
Student Details:
ID: 23
Name: Raj
Sport: Badminton
Rank: 2
```

# 🔥 Problem 4: Employee Management System

**Problem Statement:**

Create a system that stores Employee information. If the employee is a manager, also store the team size.

👉 **Hint:** Think about how "Manager" can inherit "Employee".

**Sample Input:**

```
Enter Employee ID: 1001
Enter Name: Akash
Enter Team Size: 8
```

**Expected Output:**

```
Employee ID: 1001
Name: Akash
Managing Team Size: 8
```

# 🔥 Problem 5: Product Discount System

**Problem Statement:**

Create a system that stores product information and calculates the final price after applying a discount.

**Sample Input:**

```
Enter Product ID: 501
Enter Name: Mobile
Enter Price: 15000
Enter Discount (%): 10
```

**Expected Output:**

```
Product ID: 501
Name: Mobile
Price before discount: 15000
```

Discount: 10%
Final Price: 13500

# 🔥 Problem 6: University Faculty Research Details

**Problem Statement:**

Create a system that stores Faculty information along with Research Area.

👉 **Hint:** Think about combining Faculty and Researcher properties.

**Sample Input:**

Enter Name: Dr. Shreya
Enter Age: 40
Enter Faculty ID: F123
Enter Department: Computer Science
Enter Research Area: Artificial Intelligence

**Expected Output:**

Research Faculty Details:
Name: Dr. Shreya
Age: 40
Faculty ID: F123
Department: Computer Science
Research Area: Artificial Intelligence

# 🔥 Problem 7: Online Shopping Order System

**Problem Statement:**

Create a system that takes customer order details and calculates the total bill based on product price and quantity.

**Sample Input:**

Enter Customer Name: Rohan
Enter Product Name: Laptop

```
Enter Price: 60000
Enter Quantity: 2
```

**Expected Output:**

```
Order Summary:
Customer: Rohan
Product: Laptop
Unit Price: 60000
Quantity: 2
Total Amount: 120000
```

# 🔥 Problem 8: Citizen Registration System

**Problem Statement:**

Create a system that registers a citizen and displays their ID card details.

**Sample Input:**

```
Enter Citizen Name: Sneha
Enter Age: 30
Enter City: Delhi
```

**Expected Output:**

```
Citizen ID Card:
Name: Sneha
Age: 30
City: Delhi
```

# 🔥 Problem 9: Vehicle Insurance System

**Problem Statement:**

Create a system that stores vehicle insurance details.

**Sample Input:**

```
Enter Vehicle Number: UP32AB1234
Enter Owner Name: Vikas
Enter Vehicle Type: Car
Enter Insurance Amount: 5000
```

**Expected Output:**

```
Insurance Details:
Vehicle Number: UP32AB1234
Owner: Vikas
Vehicle Type: Car
Insurance Amount: 5000
```

# 🔥 Problem 10: Movie Ticket Booking System

**Problem Statement:**

Create a system to book movie tickets and calculate the total price based on the number of tickets.

**Sample Input:**

```
Enter Movie Name: Avatar
Enter Customer Name: Rahul
Enter Number of Tickets: 3
Enter Price per Ticket: 250
```

**Expected Output:**

```
Ticket Details:
Movie: Avatar
Customer: Rahul
Number of Tickets: 3
Total Price: 750
```

Happy Coding!