OOP Concepts Overview

🎯 Direct Challenges

1. Create a class with properties and a method to display details

```
public class Student {
    String name;
    int age;

    void displayDetails() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}
```

```
Output:
Name: Sreevani
Age: 21
```

2. Demonstrate class instantiation and method invocation

```
public class Main {
    public static void main(String[] args) {
        Student s1 = new Student();
        s1.name = "Sreevani";
        s1.age = 21;
        s1.displayDetails();  // Method invocation
    }
}
```

```
Output:
Name: Sreevani
Age: 21
```

3. Use getters and setters to access private data members

```
public class Employee {
    private String empName;
    private double salary;

    public void setEmpName(String name) {
        empName = name;
    }

    public String getEmpName() {
        return empName;
    }

    public void setSalary(double sal) {
        salary = sal;
    }

    public double getSalary() {
        return salary;
    }
}

public class Main {
    public static void main(String[] args) {
        Employee emp = new Employee();
        emp.setEmpName("Vani");
        emp.setSalary(55000);

        System.out.println("Employee Name: " + emp.getEmpName());
        System.out.println("Salary: " + emp.getSalary());
    }
}
```

Output:

```
Employee Name: Vani
Salary: 55000.0
```

🧩 Scenario-Based Challenges

1. Build a Book class and create objects to store different book information

```java
public class Book {
    String title;
    String author;
    double price;

    void showBook() {
        System.out.println("Title: " + title);
        System.out.println("Author: " + author);
        System.out.println("Price: ₹" + price);
    }
}

public class Main {
    public static void main(String[] args) {
        Book b1 = new Book();
        b1.title = "Java Basics";
        b1.author = "Sree Vani";
        b1.price = 299.99;

        Book b2 = new Book();
        b2.title = "Spring Boot Guide";
        b2.author = "Ravi";
        b2.price = 499.99;

        b1.showBook();
        System.out.println();
        b2.showBook();
    }
}
```

Output:

```
Title: Java Basics
Author: Sree Vani
Price: ₹299.99

Title: Spring Boot Guide
Author: Ravi
Price: ₹499.99
```

2. Create a BankAccount class and show deposit and withdrawal actions

```
public class BankAccount {
    private String accountHolder;
    private double balance;

    public BankAccount(String name, double initialBalance) {
        accountHolder = name;
        balance = initialBalance;
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: ₹" + amount);
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: ₹" + amount);
        } else {
            System.out.println("Insufficient balance!");
        }
    }

    public void showBalance() {
        System.out.println("Account Holder: " + accountHolder);
        System.out.println("Balance: ₹" + balance);
    }
}

public class Main {
    public static void main(String[] args) {
        BankAccount acc = new BankAccount("Sreevani", 1000);
        acc.showBalance();
        acc.deposit(500);
        acc.withdraw(300);
        acc.withdraw(1500); // to show insufficient balance
        acc.showBalance();
    }
}
```

Output:

```
Account Holder: Sreevani
Balance: ₹1000.0
Deposited: ₹500.0
Withdrawn: ₹300.0
Insufficient balance!
Account Holder: Sreevani
Balance: ₹1200.0
```