Experiment - 2

Student Name : Payal Singroha **UID: 22BCS16566**

Branch: BE-CSE Section/Group: 626-B

Semester: 6th Date of Performance: 14/02/25

Subject Name: Java Subject Code: 22CSH-352

1. Aim: Develop a program for
   a) Easy Level: Product Class
   b) Medium Level: Library Management System
   c) Hard Level: Student Information System

2. Implementation/Code:

a) class Product {

```
    // Attributes
    private int id;
    private String name;
    private double price;

    // Constructor
    public Product(int id, String name,
double price) {
        this.id = id;
        this.name = name;
```

```
        this.price = price;
    }

    // Method to display product
details
    public void displayDetails() {
        System.out.println("Product
Details:");
        System.out.println("ID: " + id);
        System.out.println("Name: " +
name);
        System.out.println("Price: " +
price);
    }

    // Main method to test the class
    public static void main(String[]
args) {
        // Creating a Product object
        Product product = new
Product(101, "Laptop", 75000);

        // Displaying product details
```

```
        product.displayDetails();
    }
}
```

Output:

Product.java

```
class Product {
    // Attributes
    private int id;
    private String name;
    private double price;

    // Constructor
    public Product(int id, String name, double price) {
        this.id = id;
```

input

```
Product Details:
ID: 101
Name: Laptop
Price: 75000.0


...Program finished with exit code 0
Press ENTER to exit console.
```

B) class Book {

```
    protected String title;
    protected String author;
    protected double price;


    // Constructor
    public Book(String title, String
author, double price) {
        this.title = title;
```

```
        this.author = author;
        this.price = price;
    }

    // Method to display book details
    public void displayDetails() {
        System.out.println("Title: " +
title);
        System.out.println("Author: " +
author);
        System.out.println("Price: " +
price);
    }
}

// Derived class for Fiction books
class Fiction extends Book {
    public Fiction(String title, String
author, double price) {
        super(title, author, price);
    }

    @Override
```

```
        public void displayDetails() {
            System.out.println("Fiction
Book Details:");
            super.displayDetails();
        }
    }

    // Derived class for Non-Fiction
books
    class NonFiction extends Book {
        public NonFiction(String title,
String author, double price) {
            super(title, author, price);
        }

        @Override
        public void displayDetails() {
            System.out.println("Non-
Fiction Book Details:");
            super.displayDetails();
        }
    }
```

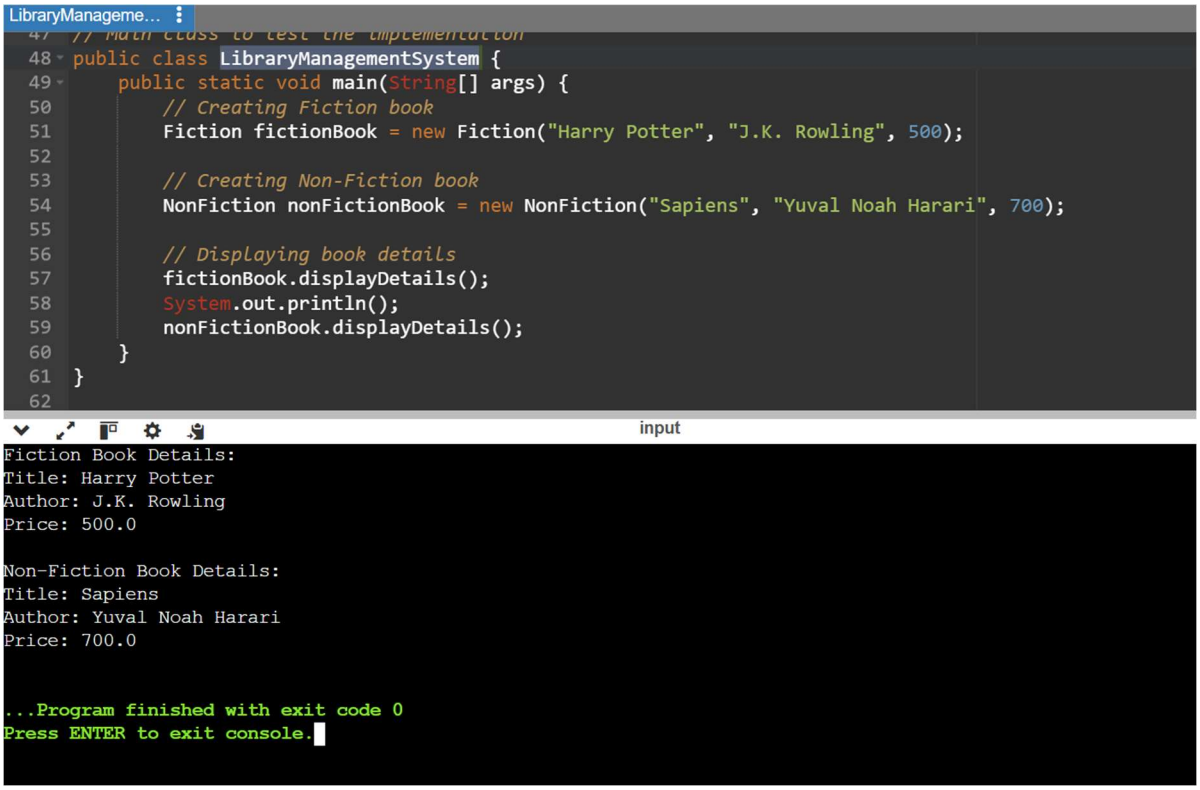
```
    // Main class to test the
implementation
    public class
LibraryManagementSystem {
        public static void main(String[]
args) {
            // Creating Fiction book
            Fiction fictionBook = new
Fiction("Harry Potter", "J.K. Rowling",
500);

            // Creating Non-Fiction book
            NonFiction nonFictionBook =
new NonFiction("Sapiens", "Yuval
Noah Harari", 700);

            // Displaying book details
            fictionBook.displayDetails();
            System.out.println();

nonFictionBook.displayDetails();
        }
    }
```

OUTPUT:



C) abstract class Person {

protected String name;

protected int age;

// Constructor

public Person(String name, int age) {

this.name = name;

this.age = age;

}

```
        // Abstract method to display
details
        public abstract void
displayDetails();
    }

    // Derived class for Student
    class Student extends Person {
        private int rollNumber;

        public Student(String name, int
age, int rollNumber) {
            super(name, age);
            this.rollNumber = rollNumber;
        }

        @Override
        public void displayDetails() {
            System.out.println("Student
Details:");
            System.out.println("Name: " +
name);
```

```
            System.out.println("Age: " +
age);
            System.out.println("Roll
Number: " + rollNumber);
        }
    }

    // Derived class for Teacher
    class Teacher extends Person {
        private String subject;

        public Teacher(String name, int
age, String subject) {
            super(name, age);
            this.subject = subject;
        }

        @Override
        public void displayDetails() {
            System.out.println("Teacher
Details:");
            System.out.println("Name: " +
name);
```

```
            System.out.println("Age: " +
age);
            System.out.println("Subject: " +
subject);
        }
    }

    // Main class to test the system
    public class
StudentInformationSystem {
        public static void main(String[]
args) {
            // Creating Student object
            Student student = new
Student("Alice", 20, 101);

            // Creating Teacher object
            Teacher teacher = new
Teacher("Mr. Smith", 40,
"Mathematics");

            // Displaying details
            student.displayDetails();
```

```
        System.out.println();

        teacher.displayDetails();

    }

}
```

OUTPUT:

StudentInformation...

```
1  abstract class Person {
2      protected String name;
3      protected int age;
4
5      // Constructor
6      public Person(String name, int age) {
7          this.name = name;
8          this.age = age;
9      }
10
11     // Abstract method to display details
12     public abstract void displayDetails();
13 }
14
15 // Derived class for Student
```

input

```
Student Details:
Name: Alice
Age: 20
Roll Number: 101

Teacher Details:
Name: Mr. Smith
Age: 40
Subject: Mathematics


...Program finished with exit code 0
Press ENTER to exit console.
```