Experiment 2

Student Name: Diksha   **UID:** 23BCS10994
Branch: CSE   **Section/Group:** KRG_2B
Semester: 5th   **Date of Performance:** 13/08/25
Subject Name: PBLJ   **Subject Code:** 23CSH-304

1. Aim:

To design and implement Java programs for managing product details, library systems, and student information using classes, inheritance, and abstraction.

A) Easy Level:
   ·   To create a Product class with attributes and constructors, and display product details.

B) Medium Level:
   ·   To implement a library management system using a base class Book and derived classes Fiction and NonFiction.

C) Hard Level:
   ·   To design a student information system using abstraction with an abstract class Person, and subclasses Student and Teacher.

2. Objective:

- To understand the use of classes, objects, constructors, and methods in Java.
- To apply object-oriented concepts for modeling real-world entities like products, books, students, and teachers.
- To demonstrate inheritance by extending a base class (Book) into derived classes (Fiction and NonFiction).
- To implement dynamic method invocation (runtime polymorphism) through method overriding in subclasses.
- To apply abstraction using an abstract class (Person) and enforce implementation of abstract methods in derived classes.

3. JAVA script and output:

EASY-LEVEL PROBLEM

```java
import java.util.Scanner;

class Product {
    int id;
    String name;
    double price;

    Product(int id, String name, double price)
    {      this.id = id;       this.name = name;
this.price = price;
    }

    void displayDetails() {
        System.out.println("Product Details:");
        System.out.println("ID: " + id);
        System.out.println("Name: " + name);
        System.out.println("Price: " + price);
    }
}

public class ProductDemo {    public static
void main(String[] args) {       Scanner sc
= new Scanner(System.in);
System.out.print("Product ID: ");        int
id = sc.nextInt();        sc.nextLine();
System.out.print("Name: ");
     String name = sc.nextLine();
System.out.print("Price: ");
     double price = sc.nextDouble();

     Product p = new Product(id, name, price);
p.displayDetails();
}
}
```

Output:

```
Product ID: 234
Name: Rice
Price: 230
Product Details:
ID: 234
Name: Rice
Price: 230.0
BUILD SUCCESSFUL (total time: 20 seconds)
```

MEDIUM LEVEL PROBLEM:

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

   Book(String title, String author, double
price) {       this.title = title;       this.author
= author;
      this.price = price;
   }

   void displayDetails() {
      System.out.println("Book Details");
   }
}

class Fiction extends Book {
   Fiction(String title, String author, double price) {
      super(title, author, price);
   }

   void displayDetails() {
      System.out.println("Fiction Book Details:");
      System.out.println("Title: " + title);
      System.out.println("Author: " + author);
      System.out.println("Price: " + price);
   }
```

```java
}

class NonFiction extends Book {
    NonFiction(String title, String author, double price) {
        super(title, author, price);
    }

    void displayDetails() {
        System.out.println("Non-Fiction Book Details:");
        System.out.println("Title: " + title);
        System.out.println("Author: " + author);
        System.out.println("Price: " + price);
    }
}

public class LibrarySystem {    public
static void main(String[] args) {

    Fiction f = new Fiction("Harry Potter", "J.K. Rowling", 500);

    NonFiction nf = new NonFiction("A Room on the Roof ", "Ruskin Bond", 700);


    f.displayDetails();
nf.displayDetails();

    }

}
```

Output:

```
Fiction Book Details:
Title: Harry Potter
Author: J.K. Rowling
Price: 500.0
Non-Fiction Book Details:                    :
Title: A Room on the Roof
Author: Ruskin Bond
Price: 700.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

HARD LEVEL PROBLEM

```java
abstract class Person
{    String name;
   int age;

   Person(String name, int age)
{        this.name = name;
this.age = age;
   }

   abstract void displayDetails();
}

class Student extends Person {
   int rollNumber;

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

   void displayDetails() {
      System.out.println("Student Details:");
      System.out.println("Name: " + name);
      System.out.println("Age: " + age);
      System.out.println("Roll Number: " + rollNumber);
   }
}

class Teacher extends Person {
   String subject;

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

   void displayDetails() {
```

```java
        System.out.println("Teacher Details:");
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Subject: " + subject);
    }
}

public class StudentInfoSystem {     public
static void main(String[] args) {         Student s
= new Student("Alice", 20, 101);
        Teacher t = new Teacher("Mr. Smith", 40, "Mathematics");

        s.displayDetails();
        t.displayDetails();
    }
}
```

Output:

```
Student Details:
Name: Diksha
Age: 20
Roll Number: 101
Teacher Details:
Name: Mr. A
Age: 40
Subject: Mathematics


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