AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF UGC ACT, 1956
SCHOOL OF COMPUTING

SRINIVAS R

CH.SC.U4CSE24146

OBJECT ORIENTED PROGRAMMING

(23CSE111)

LAB RECORD

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING, CHENNAI

BONAFIDE CERTIFICATE

This is to certify that the Lab Record work for 23CSE111- Object Oriented Programming Subject submitted by *CH.SC.U4CSE24146 – SRINIVAS R* in “Computer Science and Engineering” is a Bonafide record of the work carried out under my guidance and supervision at Amrita School of Computing, Chennai.

This Lab examination held on

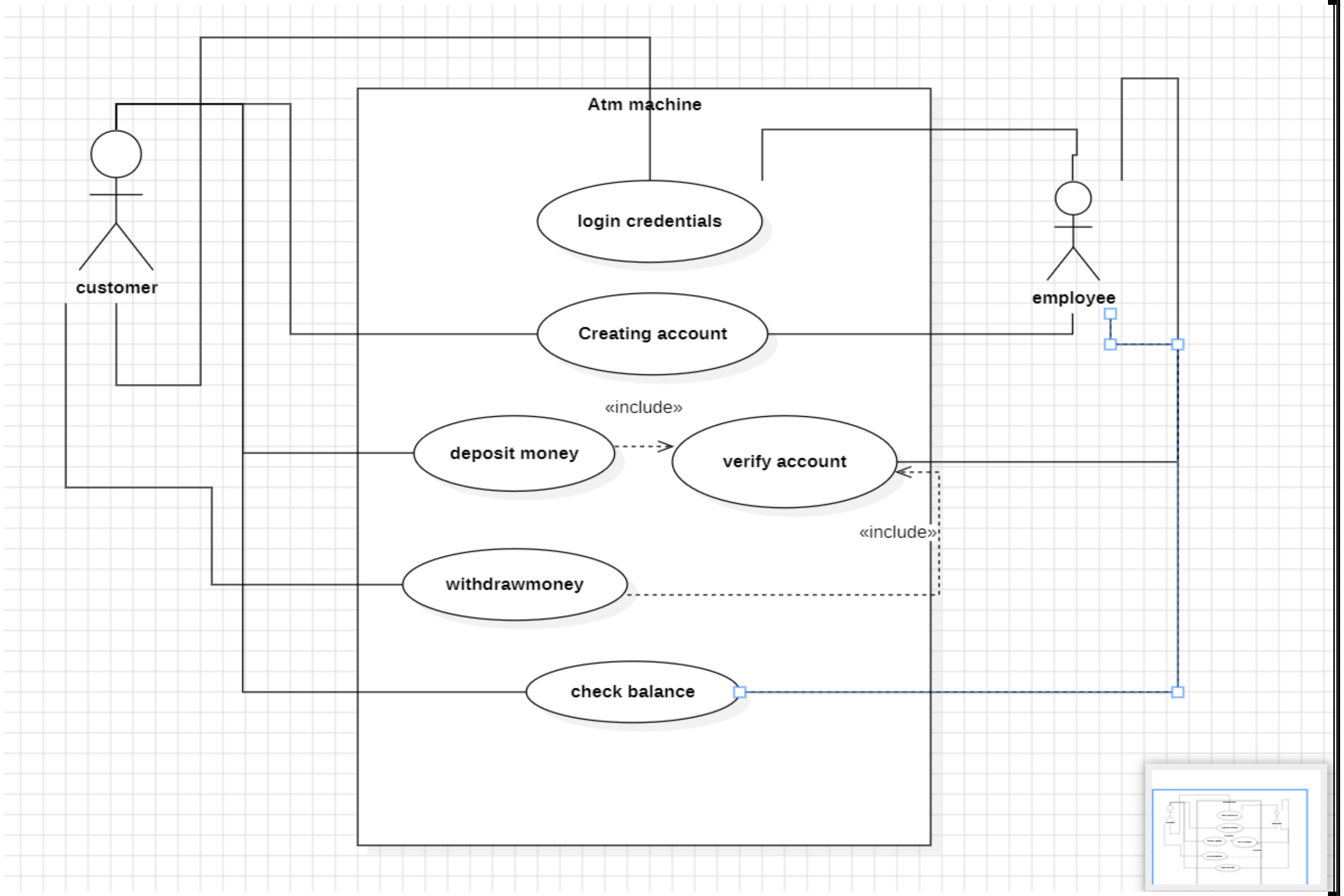
Internal Examiner 1 Internal Examiner 2

INDEX

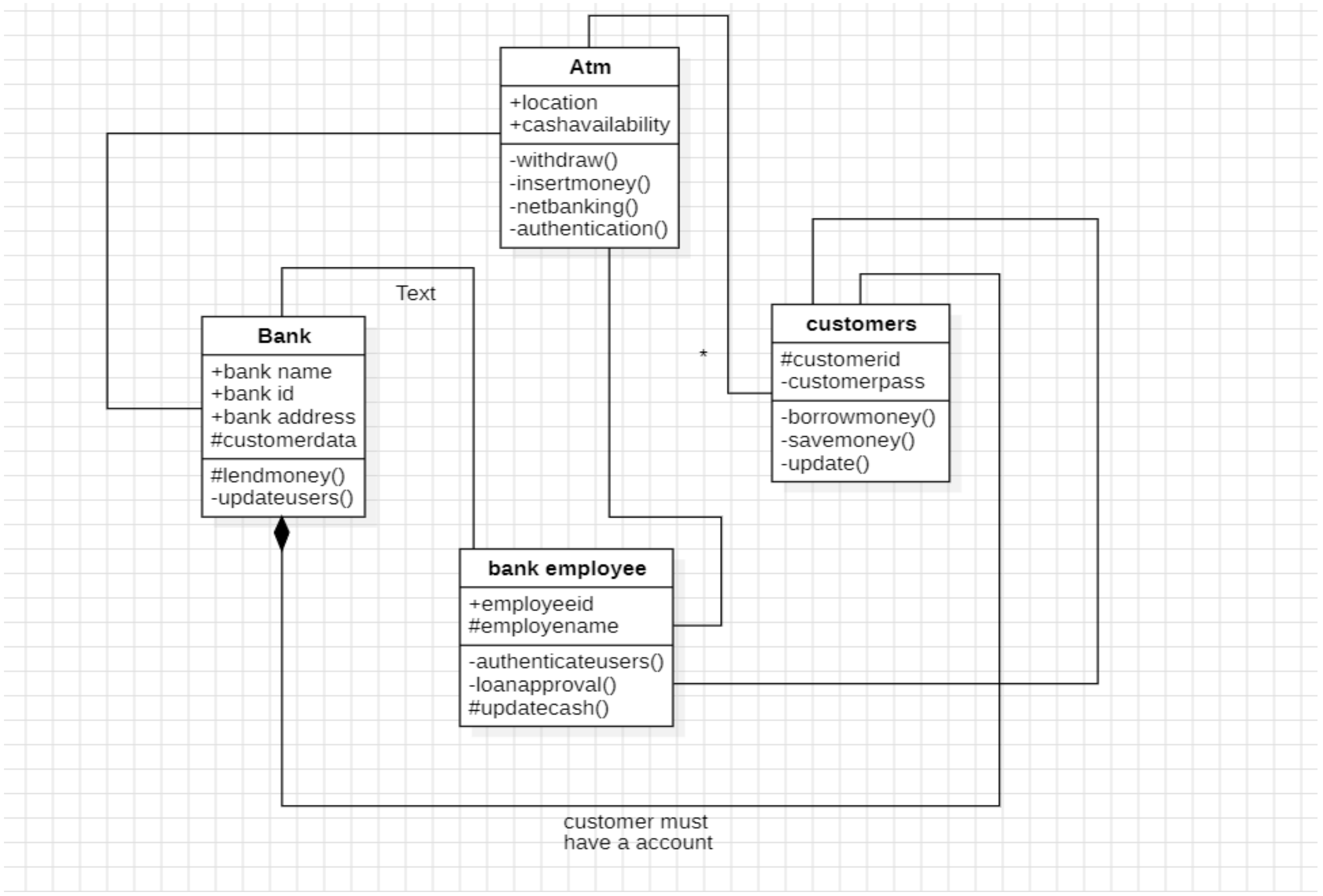
UML DIAGRAMS

1. ATM Machine

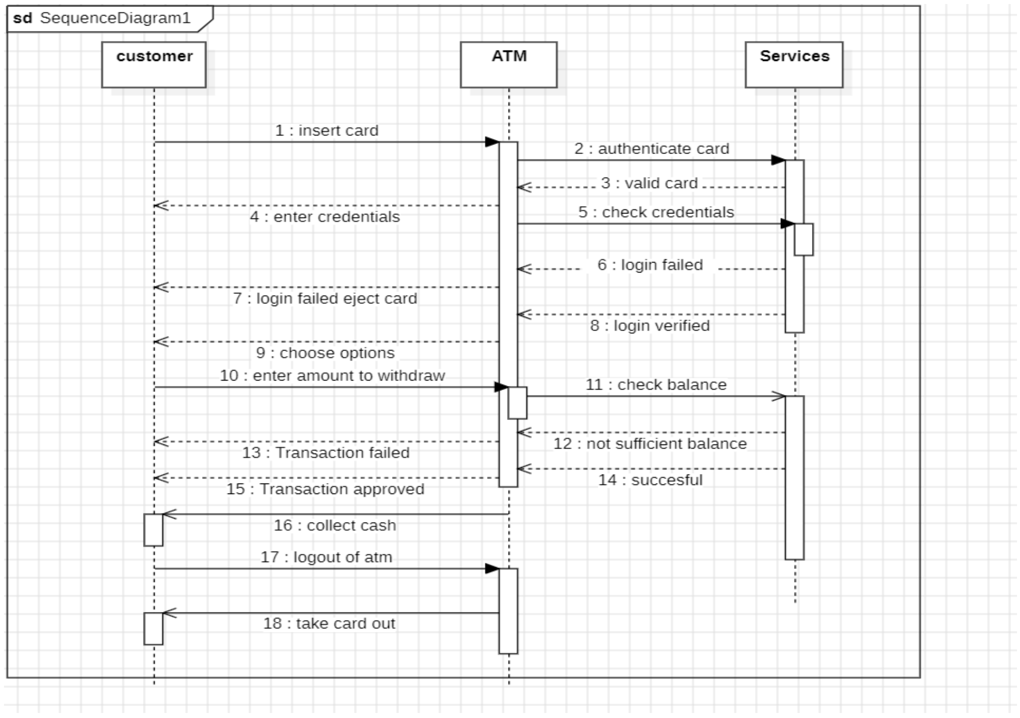
1.a) Use Case Diagram:



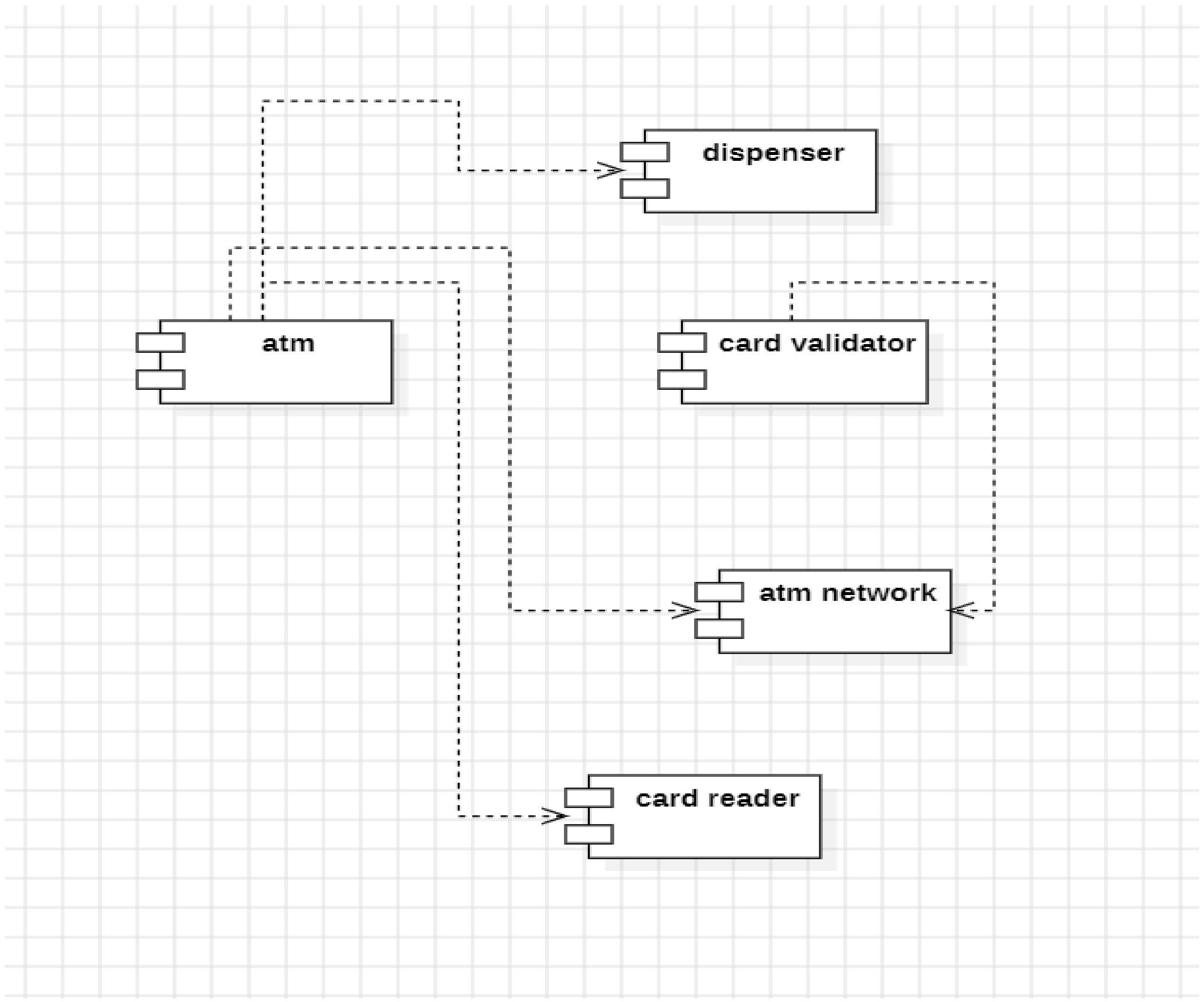
1.b) Class Diagram:



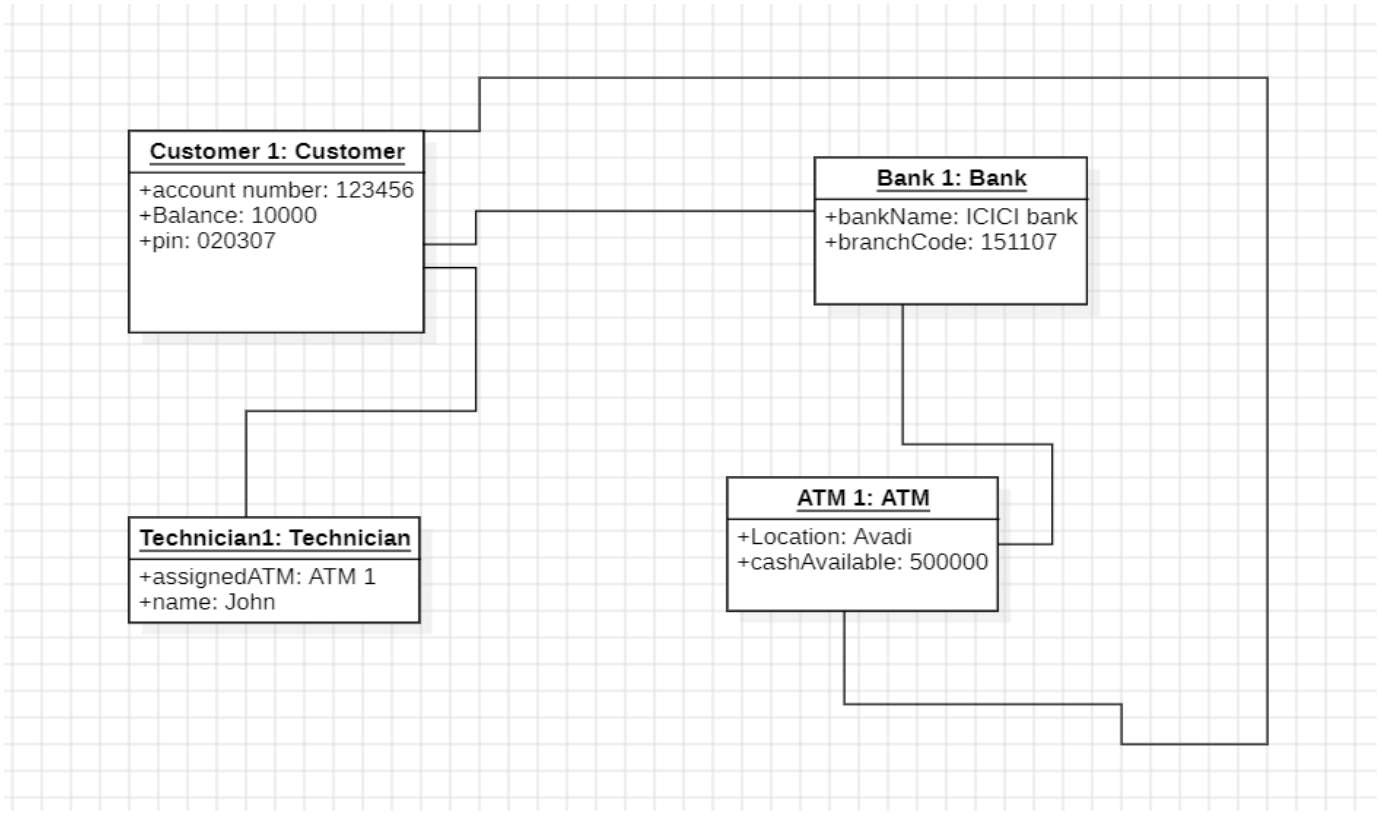
1.c) Sequence Diagram:



1.d) component diagram:

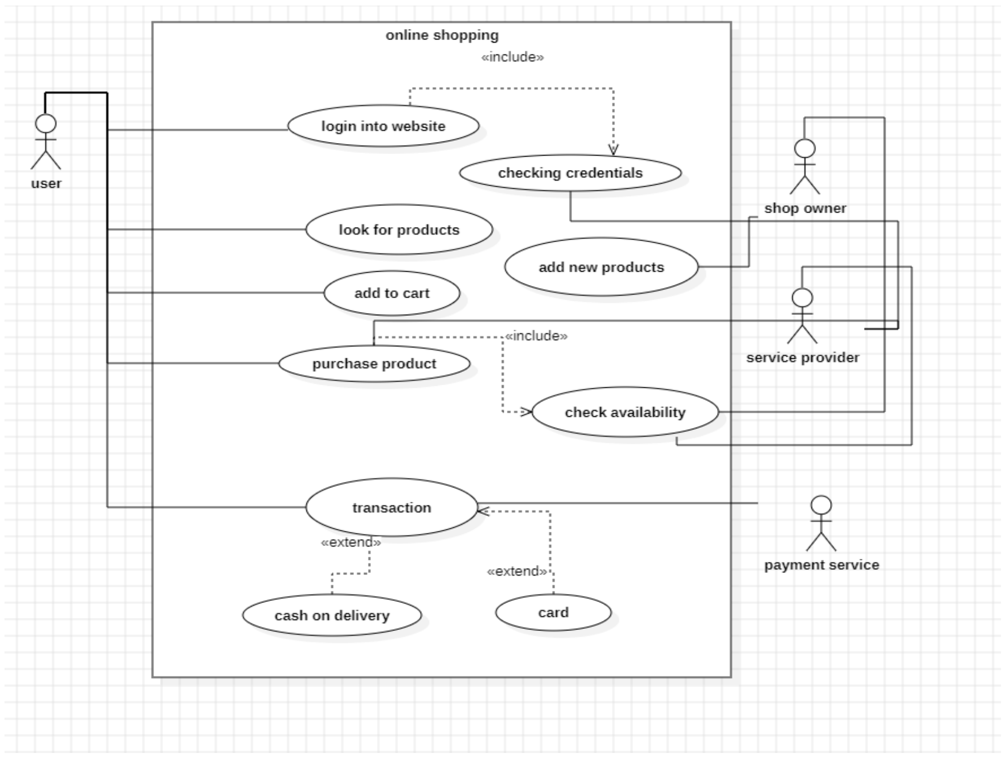


1.e) object Diagram:

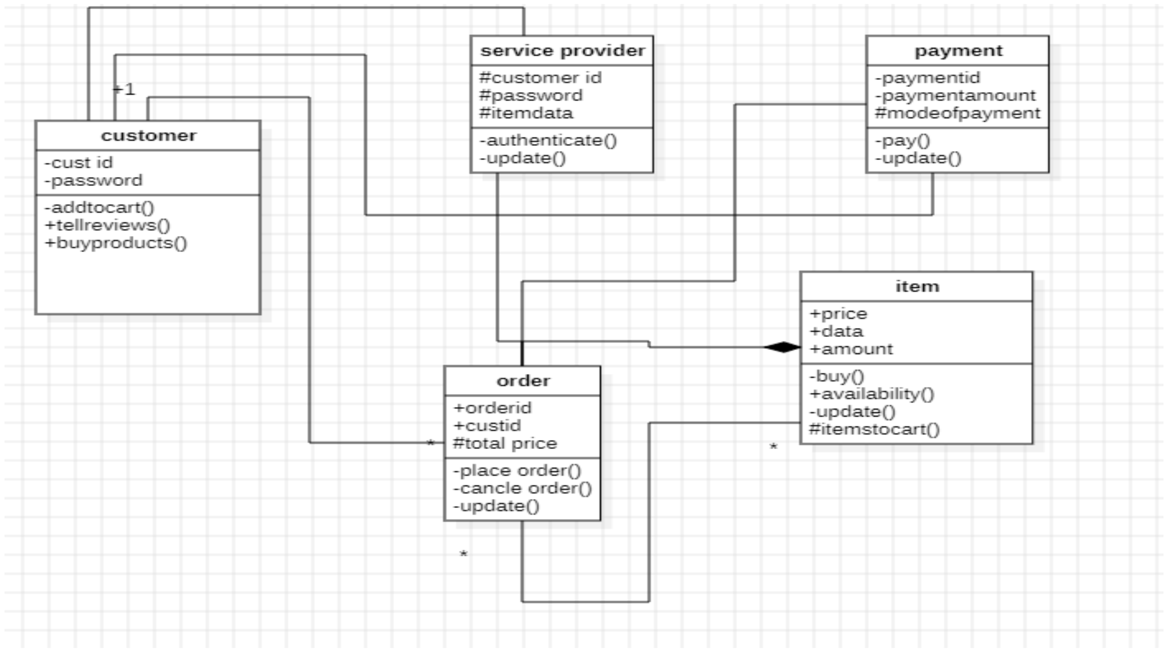


2. Shopping System

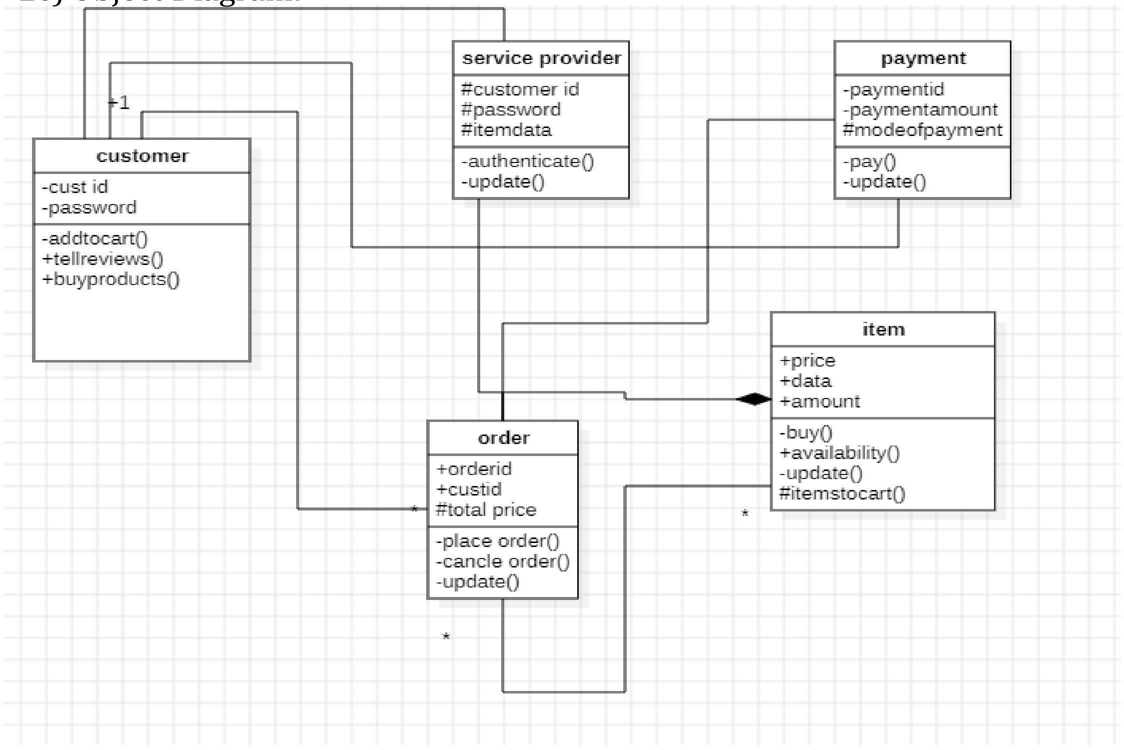
2.a) Use Case Diagram:



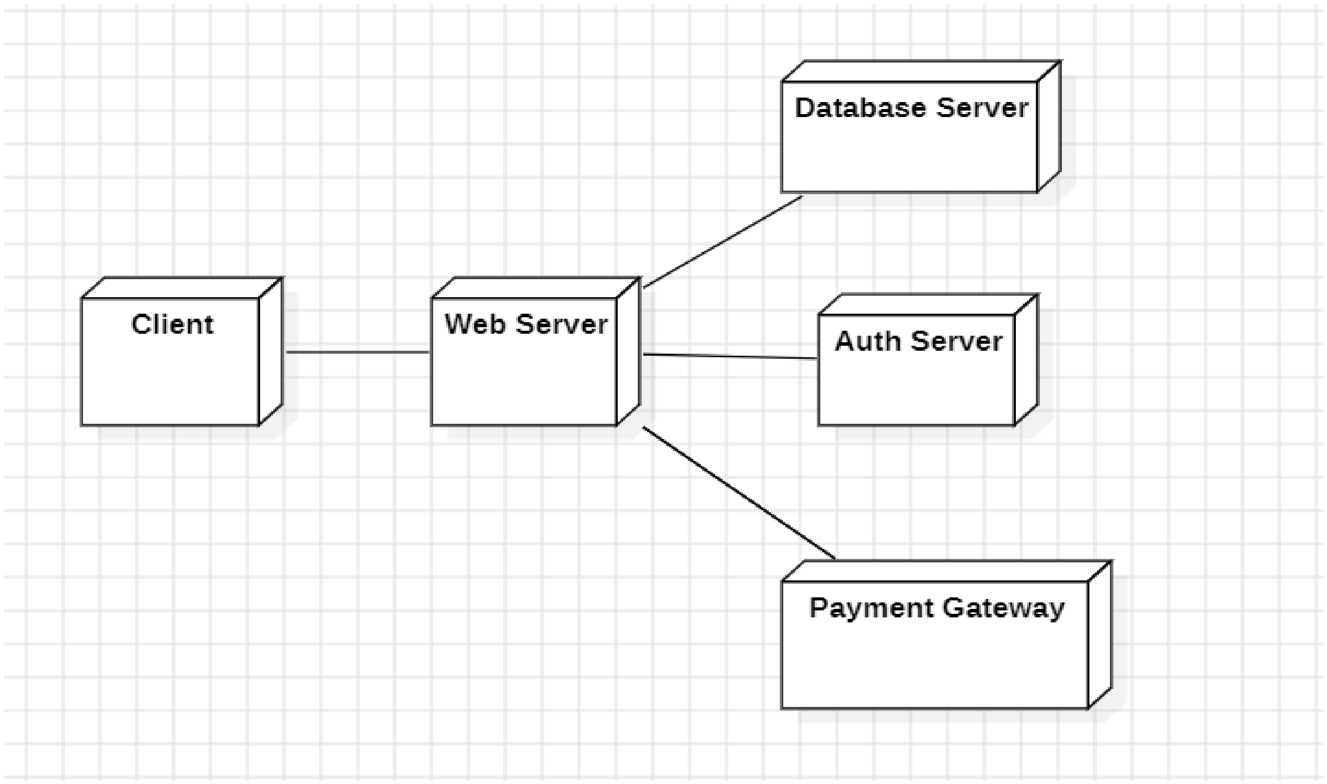
2b) Class Diagram:



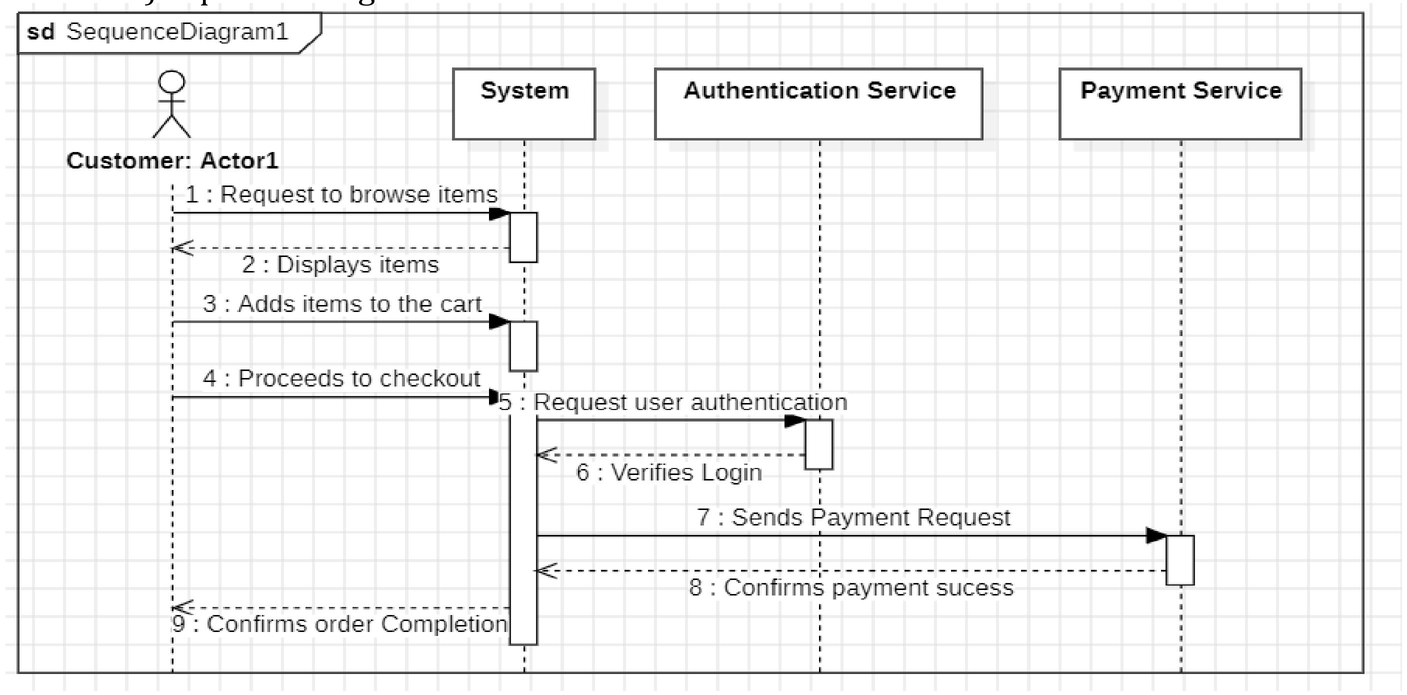
2c) Object Diagram:



2d)Deployment Diagram:



2e)Sequence Diagram:


sd SequenceDiagram1
System
Authentication Service
Payment Service
Customer: Actor1
1 : Request to browse items
2 : Displays items
3 : Adds items to the cart
4 : Proceeds to checkout
5 : Request user authentication
6 : Verifies Login
7 : Sends Payment Request
8 : Confirms payment sucess
9 : Confirms order Completion

S

Basic Java Questions

Code:

3. a) Even or odd

```
import java.util.Scanner;
public class evenodd{
public static void main(String[] args){
Scanner obj= new Scanner(System.in);
System.out.println("enter your number");
int num= obj.nextInt();
if(num%2==0){
System.out.println("the number"+num+"is even");
}
else{
System.out.println("the number"+num+"is odd");
}
obj.close();
}}
```

Output:

3.b) Student Grading

```
import java.util.Scanner;
public class Grade {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter marks: ");
        int a = scanner.nextInt();

        if (a >= 90) {
            System.out.println("Grade: A");
        } else if (a >= 80) {
            System.out.println("Grade: B");
        } else if (a >= 70) {
            System.out.println("Grade: C");
        } else if (a >= 60) {
            System.out.println("Grade: D");
        } else {
            System.out.println("Grade: F");
        }
        scanner.close();
    }
```

3.d) Simple interest calculator

```
import java.util.Scanner;
public class SimpleInterest{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Principal amount: ");
        float principal = sc.nextFloat();
        System.out.print("Enter Rate of Interest (%): ");
        float rate = sc.nextFloat();
        System.out.print("Enter Time (in years): ");
        float time =sc.nextFloat();
        float Interest = (principal * rate * time) / 100;
        System.out.println("Interest: " + Interest);
        System.out.println("Total Amount: " + (principal + Interest))
        sc.close();
    }
}
```

Output:

3i) Sum Of N Natural Numbers:

3.e) Largest Number Calculator

```
import java.util.Scanner;
public class Largest{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter three numbers: ");
        int a = sc.nextInt();
int b = sc.nextInt();
 int c = sc.nextInt();

        if (a > b && a > c) {
            System.out.println(a + " is the largest.");
        } else if (b > c) {
            System.out.println(b + " is the largest.");
        } else {
            System.out.println(c + " is the largest.");
        }
        sc.close();
    }
}
```

OUTPUT:

3i) Sum Of N Natural Numbers:

3.f) Sum of numbers

```
import java.util.Scanner;
public class sumnum{
static int n1;
static int n2;
static int n3;
public static void main(String[] args){
Scanner obj = new Scanner(System.in);
System.out.println("enter first num");
n1=obj.nextInt();
System.out.println("enter second num");
n2=obj.nextInt();
System.out.println("enter third num");
n3=obj.nextInt();
int n4=n1+n2+n3;
System.out.println("the sum of three numbers is"+ n4);
}
}
```

OUTPUT:

3j) Sum of Digits:

Code:

```
3.g) To find the cube of a number
import java.util.Scanner;
public class cube{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("enter a number");
        int a = scanner.nextInt();
System.out.println("Cube of "+a);
System.out.println(a*a*a);
        }
}
```

OUTPUT:

3j) Sum of Digits:

Code:

3.h) Reverse String

```
import java.util.Scanner;

import java.io.*;

public class reverse{

public static void main(String[] args){

Scanner obj = new Scanner(System.in);

System.out.println("enter your word");

String name= obj.nextLine();

String rev="";

for(int i=name.length()-1;i>=0;i--){

rev+=name.charAt(i);

}

System.out.println(rev);

}

}
```

OUTPUT:

3j) Sum of Digits:

3.i) To check positive or negative

```
import java.util.Scanner;
public class Numcheck{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("to check num ");
        int a = scanner.nextInt();

        if (a >0) {
            System.out.println("positive");
        } else if (a<0) {
            System.out.println("Negative");
        } else if (a==0) {
            System.out.println("zero");
        }
        else {
            System.out.println("unknown entry");
        }
    }
}
```

OUTPUT:

3j) Sum of Digits:

3.jsss) Voting eligibility

```
 import java.util.Scanner; public
class vote{
public static void main(String[] args){
Scanner obj = new Scanner(System.in);
System.out.println("enter your age"); int age
= obj.nextInt();
if(age<18){
System.out.println("not eligible to vote");
}
else{
System.out.println("eligible to vote");
}}}
```

OUTPUT:

INHERITANCE

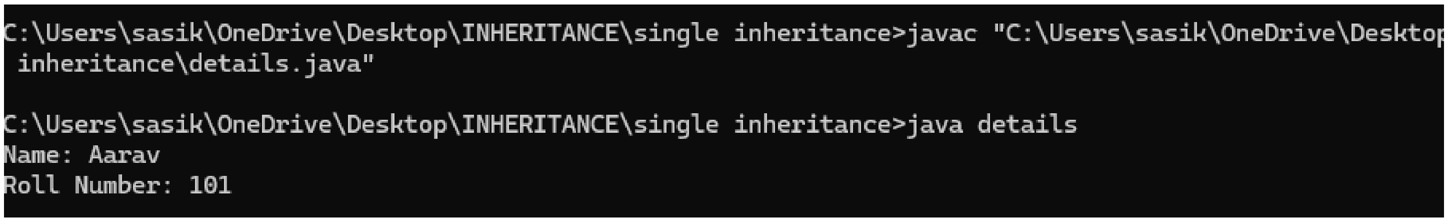
4.SINGLE INHERITANCE

4a) Students Details

Code:

```
class Person {
    String name;

    void displayInfo() {
        System.out.println("Name: " + name);
    }
}

class Student extends Person {
    int rollNumber;

    void showDetails() {
        System.out.println("Roll Number: " + rollNumber);
    }
}

public class details {
    public static void main(String[] args) {
        Student s = new Student();
        s.name = "Aarav";
        s.rollNumber = 101;

        s.displayInfo();
        s.showDetails();
    }
}
```

Screen Shot:



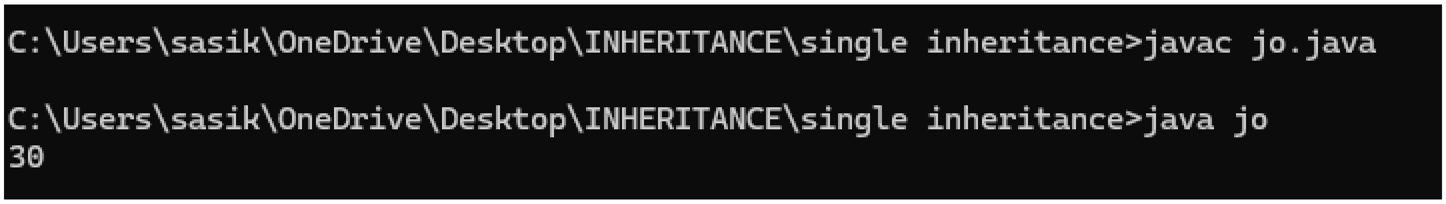
```
C:\Users\sasik\OneDrive\Desktop\INHERITANCE\single inheritance>javac "C:\Users\sasik\OneDrive\Desktop
 inheritance\details.java"

C:\Users\sasik\OneDrive\Desktop\INHERITANCE\single inheritance>java details
Name: Aarav
Roll Number: 101
```

4b) Shapes

Code:

```
class Shape{
int area;
}

class rectangle extends Shape{

rectangle(int a, int b){
area=a*b;
System.out.println(area);
}}

public class jo{
public static void main(String[] args){
rectangle obj = new rectangle(5,6);
}}
```

SCREENSHOT:



```
C:\Users\sasik\OneDrive\Desktop\INHERITANCE\single inheritance>javac jo.java

C:\Users\sasik\OneDrive\Desktop\INHERITANCE\single inheritance>java jo
30
```

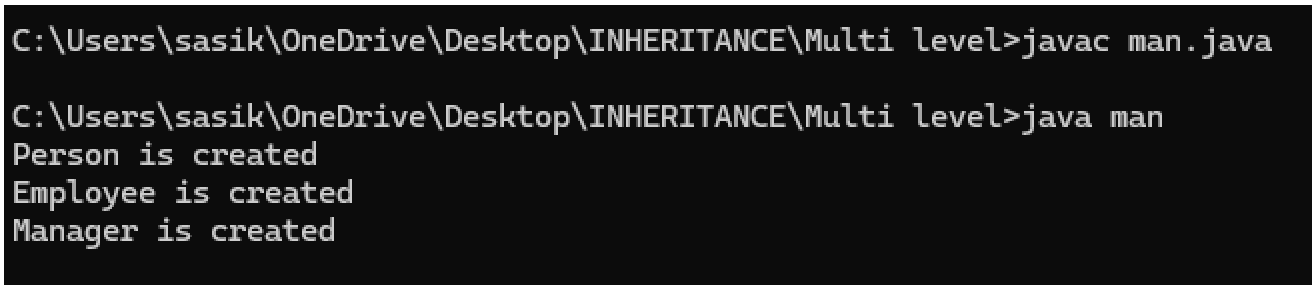
5.MULTI LEVEL INHERITANCE

5a) Employees details

CODE:

```
class Person {
    Person() {
        System.out.println("Person is created");
    }
}

class Employee extends Person {
    Employee() {
        System.out.println("Employee is created");
    }
}

class Manager extends Employee {
    Manager() {
        System.out.println("Manager is created");
    }
}

public class man {
    public static void main(String[] args) {
        Manager mgr = new Manager();
    }
}
```

SCREENSHOT:



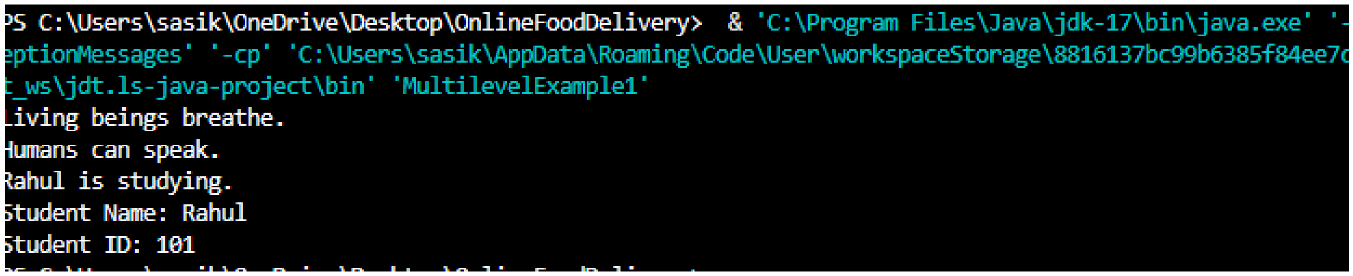
```
C:\Users\sasik\OneDrive\Desktop\INHERITANCE\Multi level>javac man.java

C:\Users\sasik\OneDrive\Desktop\INHERITANCE\Multi level>java man
Person is created
Employee is created
Manager is created
```

5b) Student details

CODE:

```
class LivingBeing {
    void breathe() {
        System.out.println("Living beings breathe.");
    }
}
class Human extends LivingBeing {
    void speak() {
        System.out.println("Humans can speak.");
    }
}
class Student extends Human {
    String name;
    int studentID;

    Student(String name, int studentID) {
        this.name = name;
        this.studentID = studentID;
    }

    void study() {
        System.out.println(name + " is studying.");
    }

    void showDetails() {
        System.out.println("Student Name: " + name);
        System.out.println("Student ID: " + studentID);
    }
}
public class MultilevelExample1 {
    public static void main(String[] args) {
        Student s1 = new Student("Rahul", 101);
        s1.breathe();
        s1.speak();
        s1.study();
        s1.showDetails();
    }
}
```

Screenshot:



```
PS C:\Users\sasik\OneDrive\Desktop\OnlineFoodDelivery>  & 'C:\Program Files\Java\jdk-17\bin\java.exe' '-
eptionMessages' '-cp' 'C:\Users\sasik\AppData\Roaming\Code\User\workspaceStorage\8816137bc99b6385f84ee7c
t_ws\jdt.ls-java-project\bin' 'MultilevelExample1'
Living beings breathe.
Humans can speak.
Rahul is studying.
Student Name: Rahul
Student ID: 101
```

6.HIERARCHICAL INHERITANCE

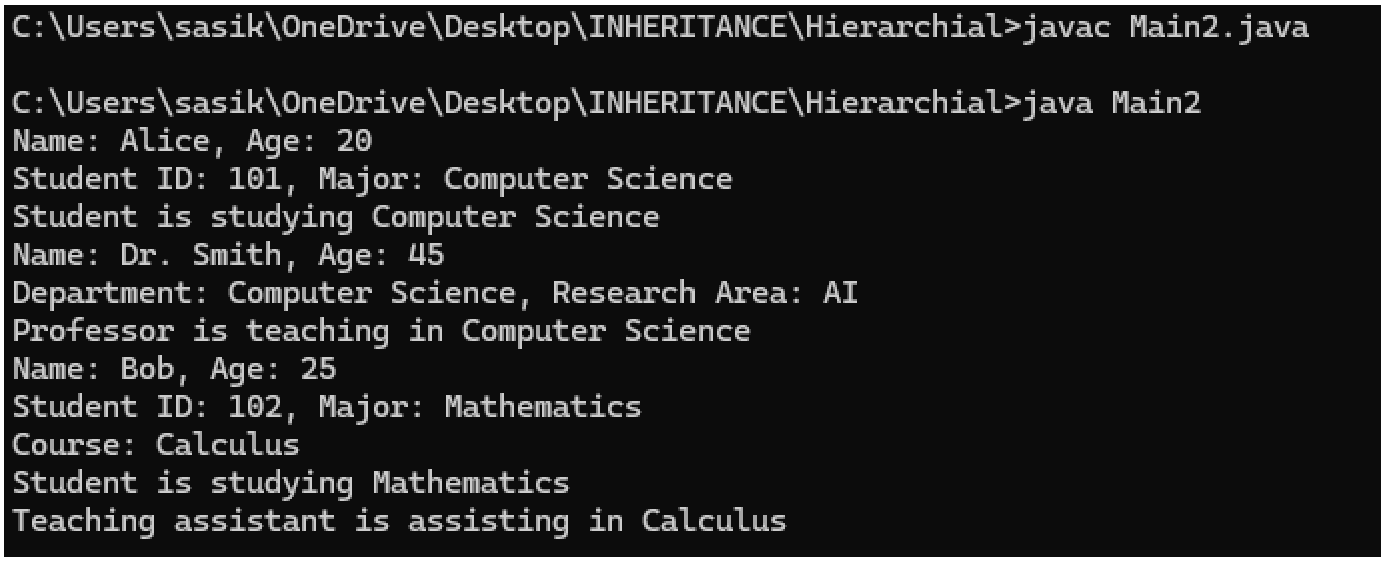
6a) Student Personal

CODE:

```
class Person {
    private String name;
    private int age;

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

    public void displayDetails() {
        System.out.println("Name: " + name + ", Age: " + age);
    }
}
class Student extends Person {
    private int studentId;
    private String major;
    public Student(String name, int age, int studentId, String major) {
        super(name, age);
        this.studentId = studentId;
        this.major = major;
    }
    public void study() {
        System.out.println("Student is studying " + major);
    }
  public void displayDetails() {
        super.displayDetails();
        System.out.println("Student ID: " + studentId + ", Major: " +
major);
```

```
    }
}
class Professor extends Person {
    private String department;
    private String researchArea;
  public Professor(String name, int age, String department, String
researchArea) {
        super(name, age);
        this.department = department;
        this.researchArea = researchArea;
    }
 public void teach() {
        System.out.println("Professor is teaching in " + department);
    }
   public void displayDetails() {
        super.displayDetails();
        System.out.println("Department: " + department + ", Research Area:
" + researchArea);
    }
}
class TeachingAssistant extends Student {
    private String course;

    public TeachingAssistant(String name, int age, int studentId, String
major, String course) {
        super(name, age, studentId, major);
        this.course = course;
    }
    public void assist() {
        System.out.println("Teaching assistant is assisting in " + course);
    }
public void displayDetails() {
        super.displayDetails();
        System.out.println("Course: " + course);
    }
}
public class Main2 {
    public static void main(String[] args) {
        Student student = new Student("Alice", 20, 101, "Computer
Science");
        student.displayDetails();
        student.study();
         Professor professor = new Professor("Dr. Smith", 45, "Computer
Science", "AI");
        professor.displayDetails();
        professor.teach();
        TeachingAssistant ta = new TeachingAssistant("Bob", 25, 102,
"Mathematics", "Calculus");
        ta.displayDetails();
        ta.study();
        ta.assist();
    }
}
```

SCREENSHOT:



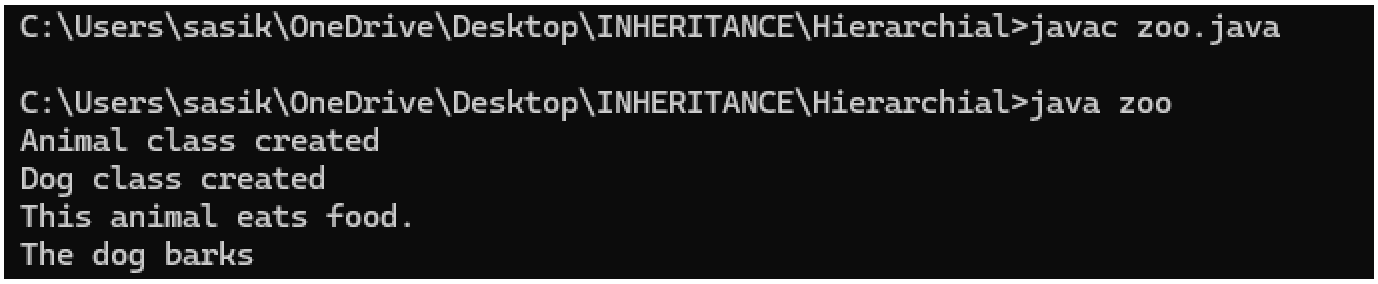
```
C:\Users\sasik\OneDrive\Desktop\INHERITANCE\Hierarchial>javac Main2.java

C:\Users\sasik\OneDrive\Desktop\INHERITANCE\Hierarchial>java Main2
Name: Alice, Age: 20
Student ID: 101, Major: Computer Science
Student is studying Computer Science
Name: Dr. Smith, Age: 45
Department: Computer Science, Research Area: AI
Professor is teaching in Computer Science
Name: Bob, Age: 25
Student ID: 102, Major: Mathematics
Course: Calculus
Student is studying Mathematics
Teaching assistant is assisting in Calculus
```

6b) ZOO

CODE:

```
class Animal {
    public Animal() {
        System.out.println("Animal class created");
    }

    public void eat() {
        System.out.println("This animal eats food.");
    }
}

class Dog extends Animal {
    public Dog() {
        System.out.println("Dog class created");
    }

    public void bark() {
        System.out.println("The dog barks");
    }
}

class Cat extends Animal {
    public Cat() {
        System.out.println("Cat class created");
    }

    public void meow() {
        System.out.println("The cat meows");
    }
}

public class zoo {
    public static void main(String[] args) {
        Dog dog = new Dog();
        dog.eat();
        dog.bark();

        System.out.println();

        Cat cat = new Cat();
        cat.eat();
        cat.meow();
    }
}
```

SCREENSHOT:



```
C:\Users\sasik\OneDrive\Desktop\INHERITANCE\Hierarchial>javac zoo.java

C:\Users\sasik\OneDrive\Desktop\INHERITANCE\Hierarchial>java zoo
Animal class created
Dog class created
This animal eats food.
The dog barks
```

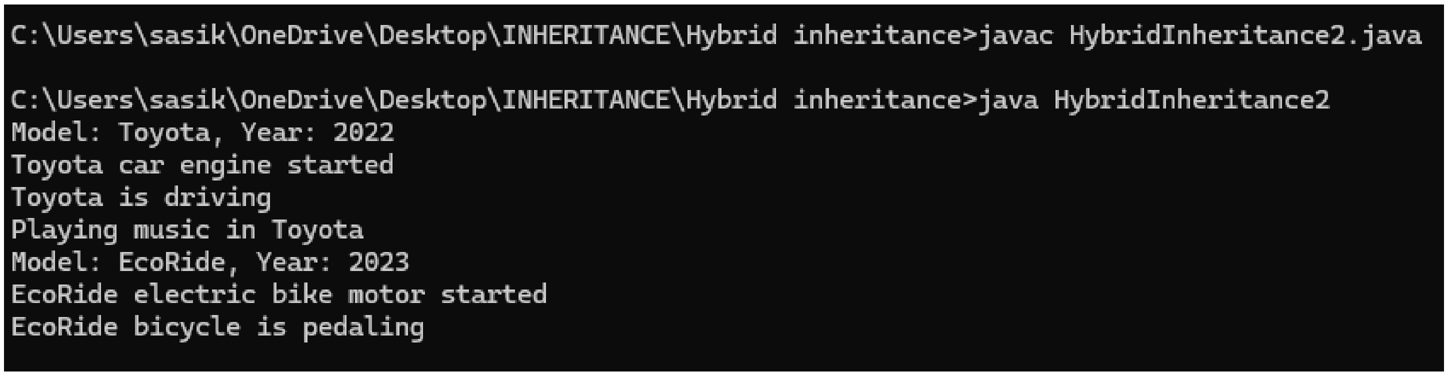
7. HYBRID INHERITANCE

7a)Vehical characteristics

CODE:

```
interface Engine {
    void startEngine();
    void stopEngine();
}

interface EntertainmentSystem {
    void playMusic();
}

class Vehicle {
    String model;
    int year;

    Vehicle(String model, int year) {
        this.model = model;
        this.year = year;
    }

    void displayInfo() {
        System.out.println("Model: " + model + ", Year: " + year);
    }
}

class Car extends Vehicle implements Engine, EntertainmentSystem {
    Car(String model, int year) {
        super(model, year);
    }

    public void startEngine() {
        System.out.println(model + " car engine started");
    }

    public void stopEngine() {
        System.out.println(model + " car engine stopped");
    }

    public void playMusic() {
        System.out.println("Playing music in " + model);
    }
    void drive() {
        System.out.println(model + " is driving");
    }
}

class Bicycle extends Vehicle {
    Bicycle(String model, int year) {
        super(model, year);
    }

    void pedal() {
        System.out.println(model + " bicycle is pedaling");
    }
}

class ElectricBike extends Bicycle implements Engine {
    ElectricBike(String model, int year) {
        super(model, year);
    }

    public void startEngine() {
        System.out.println(model + " electric bike motor started");
    }

    public void stopEngine() {
        System.out.println(model + " electric bike motor stopped");
    }
}

public class HybridInheritance2 {
    public static void main(String[] args) {
        Car myCar = new Car("Toyota", 2022);
        myCar.displayInfo();
        myCar.startEngine();
        myCar.drive();
        myCar.playMusic();

        ElectricBike myBike = new ElectricBike("EcoRide", 2023);
        myBike.displayInfo();
        myBike.startEngine();
        myBike.pedal();
    }
}
```

SCREENSHOT:

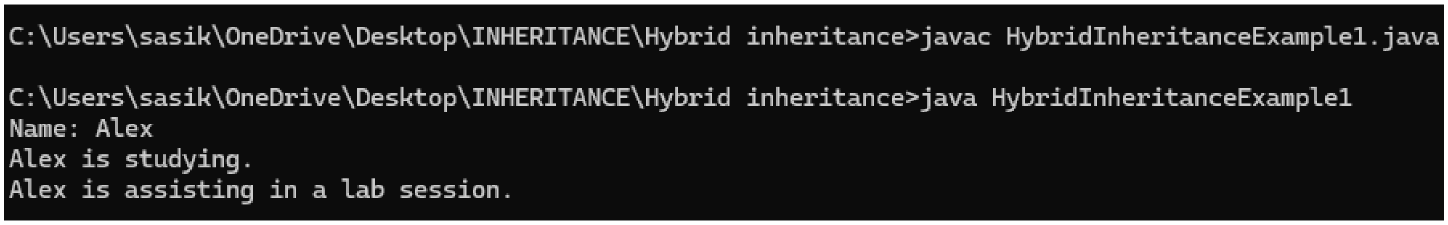

```
C:\Users\sasik\OneDrive\Desktop\INHERITANCE\Hybrid inheritance>javac HybridInheritance2.java

C:\Users\sasik\OneDrive\Desktop\INHERITANCE\Hybrid inheritance>java HybridInheritance2
Model: Toyota, Year: 2022
Toyota car engine started
Toyota is driving
Playing music in Toyota
Model: EcoRide, Year: 2023
EcoRide electric bike motor started
EcoRide bicycle is pedaling
```

7b)Details of teacher and students

CODE:

```
class Person {
    String name;

    Person(String name) {
        this.name = name;
    }

    void showDetails() {
        System.out.println("Name: " + name);
    }
}
class Student extends Person {
    int studentID;

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

    void study() {
        System.out.println(name + " is studying.");
    }
}
class Teacher extends Person {
    String subject;

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

    void teach() {
        System.out.println(name + " is teaching " + subject + ".");
    }
}
interface Assistant {
    void assist();
}
class TeachingAssistant extends Student implements Assistant {
    TeachingAssistant(String name, int studentID) {
        super(name, studentID);
    }

    public void assist() {
        System.out.println(name + " is assisting in a lab session.");
    }
}
public class HybridInheritanceExample1 {
    public static void main(String[] args) {
        TeachingAssistant ta = new TeachingAssistant("Alex", 101);
        ta.showDetails();
        ta.study();
        ta.assist();
    }
}
```

<ins>SCREENSHOT:</ins>



```
C:\Users\sasik\OneDrive\Desktop\INHERITANCE\Hybrid inheritance>javac HybridInheritanceExample1.java

C:\Users\sasik\OneDrive\Desktop\INHERITANCE\Hybrid inheritance>java HybridInheritanceExample1
Name: Alex
Alex is studying.
Alex is assisting in a lab session.
```

POLYMORPHISM

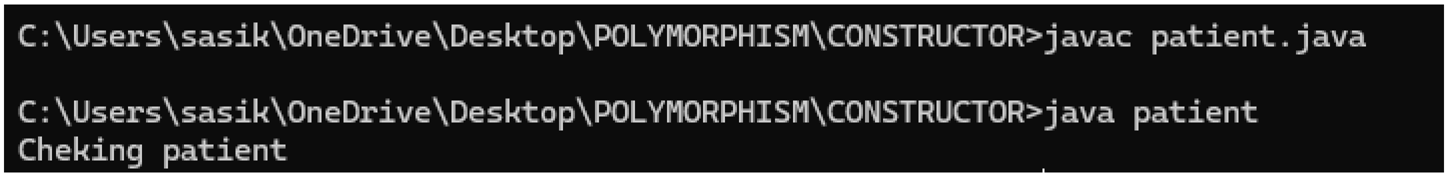
8) Constructor

a)check patient

CODE:

```
import java.util.Scanner;
public class patient{
    public static void main(String[] args) {
        Scanner obj = new Scanner(System.in);
        doctor doc = new doctor();
    }
}
class doctor{

    doctor(){
        System.out.print("Cheking patient");
    }
}
```

SCREENSHOT:



```
C:\Users\sasik\OneDrive\Desktop\POLYMORPHISM\CONSTRUCTOR>javac patient.java

C:\Users\sasik\OneDrive\Desktop\POLYMORPHISM\CONSTRUCTOR>java patient
Cheking patient
```

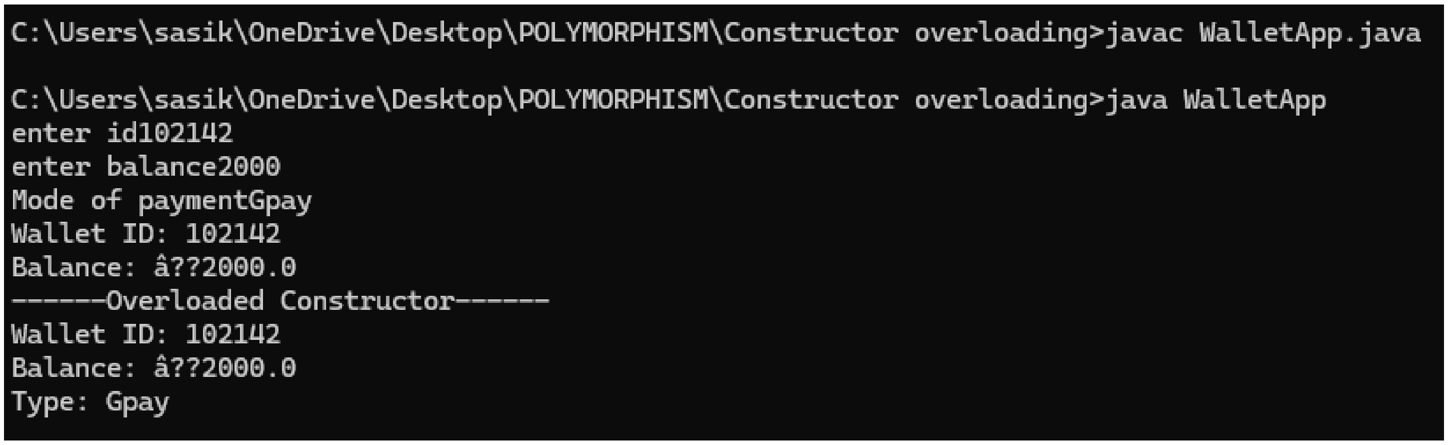
9.CONSTRUCTOR OVERLOADING

a) Wallet app

Code:

```
import java.util.Scanner;

class Wallet {
    int id;
    double balance;
    String type;

    // Constructor without type
    Wallet(int id, double balance) {
        this.id = id;
        this.balance = balance;
        System.out.println("Wallet ID: " + id);
        System.out.println("Balance: ₹" + balance);
    }

    // Constructor with type
    Wallet(int id, double balance, String type) {
        this.id = id;
        this.balance = balance;
        this.type = type;
        System.out.println("Wallet ID: " + id);
        System.out.println("Balance: ₹" + balance);
        System.out.println("Type: " + type);
    }
}

public class WalletApp {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("enter id");
        int id = sc.nextInt();
        sc.nextLine();
        System.out.print("enter balance");
        double bal = sc.nextDouble();
        sc.nextLine();
        System.out.print("Mode of payment");
        String type = sc.nextLine();

        Wallet w1 = new Wallet(id, bal);
        System.out.println("------Overloaded Constructor------");
        Wallet w2 = new Wallet(id, bal, type);

        sc.close();
    }
}
```

<u>SCREENSHOT:</u>



```
C:\Users\sasik\OneDrive\Desktop\POLYMORPHISM\Constructor overloading>javac WalletApp.java

C:\Users\sasik\OneDrive\Desktop\POLYMORPHISM\Constructor overloading>java WalletApp
enter id102142
enter balance2000
Mode of paymentGpay
Wallet ID: 102142
Balance: â??2000.0
------Overloaded Constructor------
Wallet ID: 102142
Balance: â??2000.0
Type: Gpay
```

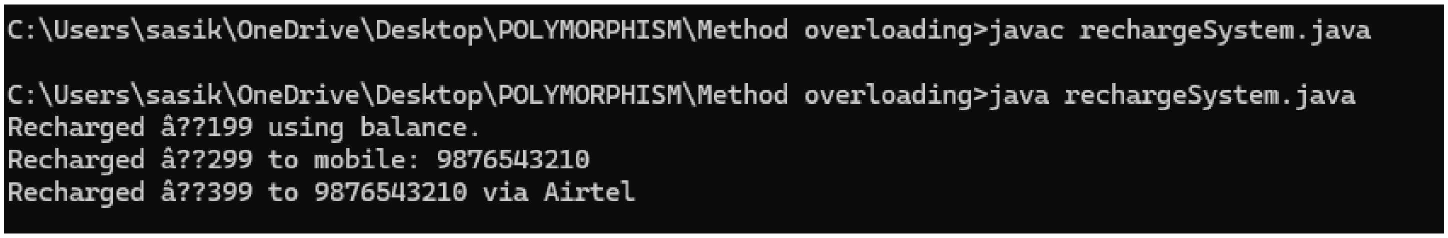
10.METHOD OVERLOADING

a)**Recharge Phone**

Code:

```
public class rechargeSystem {
    void recharge(int amount) {
        System.out.println("Recharged ₹" + amount + " using balance.");
    }

    void recharge(String mobileNumber, int amount) {
        System.out.println("Recharged ₹" + amount + " to mobile: " +
mobileNumber);
    }

    void recharge(String mobileNumber, String operator, int amount) {
        System.out.println("Recharged ₹" + amount + " to " + mobileNumber +
" via " + operator);
    }

    public static void main(String[] args) {
        rechargeSystem rs = new rechargeSystem();
        rs.recharge(199);
        rs.recharge("9876543210", 299);
        rs.recharge("9876543210", "Airtel", 399);
    }
}
```

Screen shot:



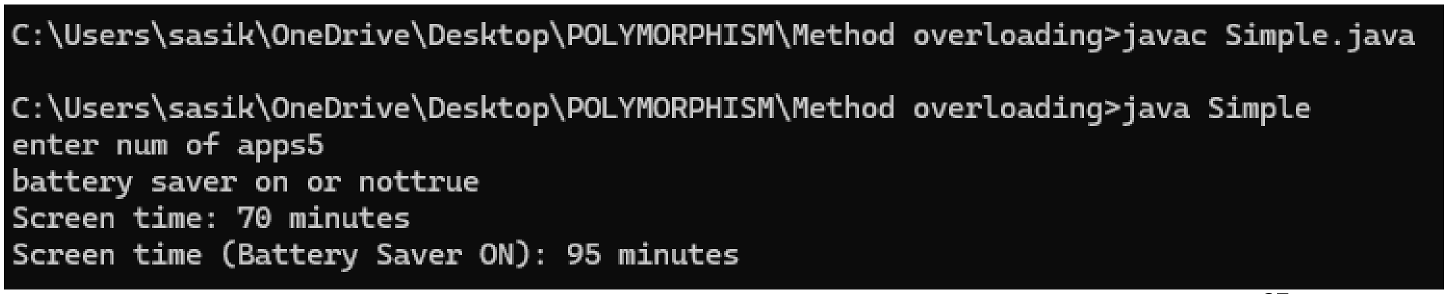
```
C:\Users\sasik\OneDrive\Desktop\POLYMORPHISM\Method overloading>javac rechargeSystem.java

C:\Users\sasik\OneDrive\Desktop\POLYMORPHISM\Method overloading>java rechargeSystem.java
Recharged â??199 using balance.
Recharged â??299 to mobile: 9876543210
Recharged â??399 to 9876543210 via Airtel
```

b) Screen timing app

CODE:

```
import java.util.Scanner;

class ScreenTime {
    void time(int apps) {
        System.out.println("Screen time: " + (120 - apps * 10) + "
minutes");
    }

    void time(int apps, boolean saver) {
        if (saver) {
            System.out.println("Screen time (Battery Saver ON): " + (120 -
apps * 5) + " minutes");
        } else {
            time(apps);
        }
    }
}

public class Simple {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("enter num of apps");
        int apps = sc.nextInt();
        System.out.print("battery saver on or not");
        boolean saver = sc.nextBoolean();

        ScreenTime s = new ScreenTime();
        s.time(apps);
        s.time(apps, saver);

        sc.close();
    }
}
```

SCREENSHOT:



```
C:\Users\sasik\OneDrive\Desktop\POLYMORPHISM\Method overloading>javac Simple.java

C:\Users\sasik\OneDrive\Desktop\POLYMORPHISM\Method overloading>java Simple
enter num of apps5
battery saver on or nottrue
Screen time: 70 minutes
Screen time (Battery Saver ON): 95 minutes
```

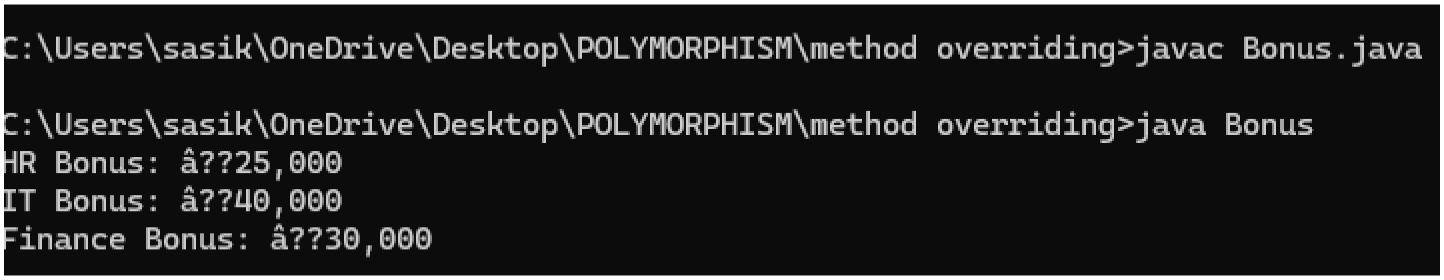
11. METHOD OVERRIDING

a) Checking Bonus

CODE:

```
public class Bonus{
    public static void main(String[] args) {
        Department d;

        d = new HR();
        d.showBonus();

        d = new IT();
        d.showBonus();

        d = new Finance();
        d.showBonus();
    }
}

class Department {
    void showBonus() {
        System.out.println("General Department Bonus");
    }
}

class HR extends Department {
    @Override
    void showBonus() {
        System.out.println("HR Bonus: ₹25,000");
    }
}

class IT extends Department {
    @Override
    void showBonus() {
        System.out.println("IT Bonus: ₹40,000");
    }
}

class Finance extends Department {
    @Override
    void showBonus() {
        System.out.println("Finance Bonus: ₹30,000");
    }
}
```

SCREENSHOT:



```
C:\Users\sasik\OneDrive\Desktop\POLYMORPHISM\method overriding>javac Bonus.java

C:\Users\sasik\OneDrive\Desktop\POLYMORPHISM\method overriding>java Bonus
HR Bonus: â??25,000
IT Bonus: â??40,000
Finance Bonus: â??30,000
```

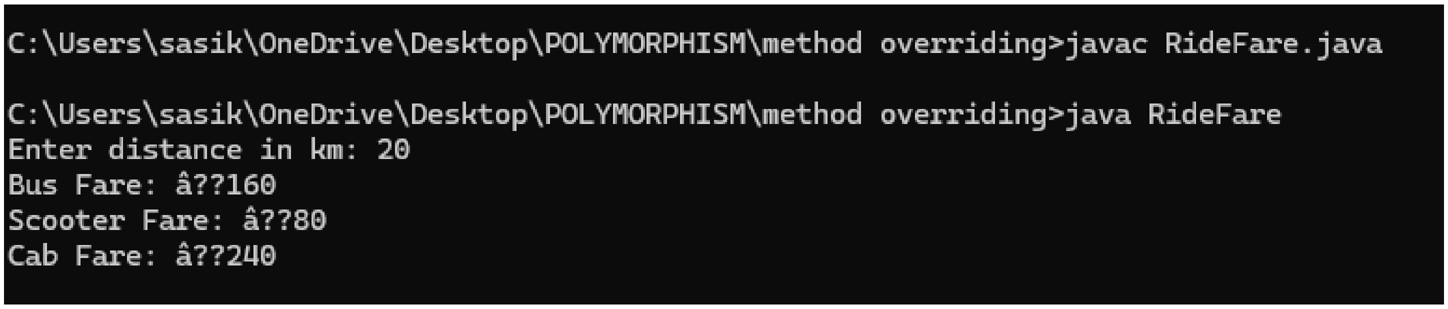
b) Riding Fare

CODE:

```
import java.util.Scanner;

class Transport {
    void showFare(int km) {
        System.out.println("Fare calculation for " + km + " km.");
    }
}

class Bus extends Transport {
    @Override
    void showFare(int km) {
        System.out.println("Bus Fare: ₹" + (km * 8));
    }
}

class Scooter extends Transport {
    @Override
    void showFare(int km) {
        System.out.println("Scooter Fare: ₹" + (km * 4));
    }
}

class Cab extends Transport {
    @Override
    void showFare(int km) {
        System.out.println("Cab Fare: ₹" + (km * 12));
    }
}
```

```
public class RideFare {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        System.out.print("Enter distance in km: ");
        int km = scan.nextInt();

        Transport t1 = new Bus();
        Transport t2 = new Scooter();
        Transport t3 = new Cab();

        t1.showFare(km);
        t2.showFare(km);
        t3.showFare(km);

        scan.close();
    }
}
```

SCREENSHOT:



```
C:\Users\sasik\OneDrive\Desktop\POLYMORPHISM\method overriding>javac RideFare.java

C:\Users\sasik\OneDrive\Desktop\POLYMORPHISM\method overriding>java RideFare
Enter distance in km: 20
Bus Fare: â??160
Scooter Fare: â??80
Cab Fare: â??240
```

12. ABSTRACTION

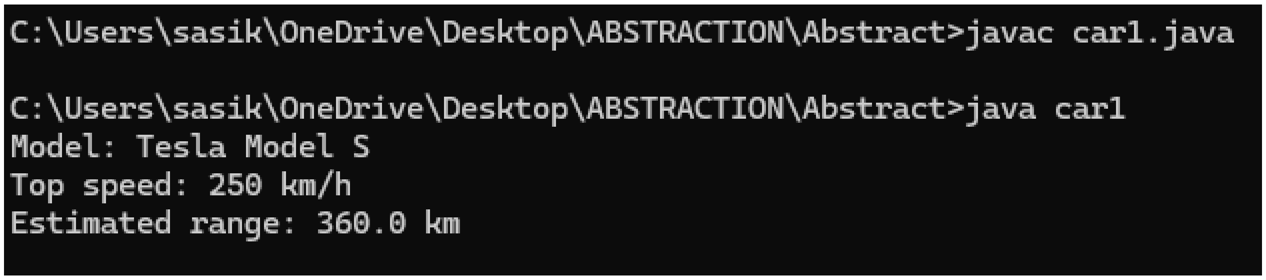
Using Abstract class

A) Car Details

CODE:

```
public class car1 {
    public static void main(String[] args) {
        Vehicle v = new Tesla();
        v.displayModel("Tesla Model S");
        v.showSpeed(250);
        v.calculateRange(50, 7.2);
    }
}

abstract class Vehicle {
    abstract void displayModel(String model);
    abstract void showSpeed(int speed);
    abstract void calculateRange(int batteryLevel, double efficiency);
}

class Tesla extends Vehicle {
    void displayModel(String model) {
        System.out.println("Model: " + model);
    }

    void showSpeed(int speed) {
        System.out.println("Top speed: " + speed + " km/h");
    }

    void calculateRange(int batteryLevel, double efficiency) {
        double range = batteryLevel * efficiency;
        System.out.println("Estimated range: " + range + " km");
    }
}
```

SCREENSHOT:



```
C:\Users\sasik\OneDrive\Desktop\ABSTRACTION\Abstract>javac car1.java

C:\Users\sasik\OneDrive\Desktop\ABSTRACTION\Abstract>java car1
Model: Tesla Model S
Top speed: 250 km/h
Estimated range: 360.0 km
```

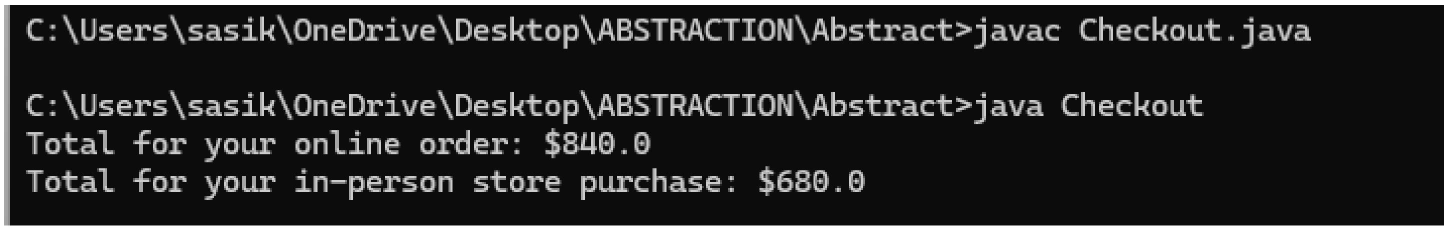
b) Online checkout

CODE:

```
abstract class Purchase {
    abstract double calculateTotalAmount();
}

class OnlinePurchase extends Purchase {
    private double itemCost, shippingCost;

    OnlinePurchase(double itemCost, double shippingCost) {
        this.itemCost = itemCost;
        this.shippingCost = shippingCost;
    }

    double calculateTotalAmount() {
        return itemCost + shippingCost;
    }
}

class InPersonPurchase extends Purchase {
    private double itemCost, discount;

    InPersonPurchase(double itemCost, double discount) {
        this.itemCost = itemCost;
        this.discount = discount;
    }

    double calculateTotalAmount() {
        return itemCost - discount; // Total = Item cost - Discount
    }
}
```

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

        Purchase onlineOrder = new OnlinePurchase(800, 40);

        Purchase inStoreOrder = new InPersonPurchase(800, 120);

        System.out.println("Total for your online order: $" +
onlineOrder.calculateTotalAmount());
        System.out.println("Total for your in-person store purchase: $" +
inStoreOrder.calculateTotalAmount());
    }
}
```

SCREENSHOT:



```
C:\Users\sasik\OneDrive\Desktop\ABSTRACTION\Abstract>javac Checkout.java

C:\Users\sasik\OneDrive\Desktop\ABSTRACTION\Abstract>java Checkout
Total for your online order: $840.0
Total for your in-person store purchase: $680.0
```

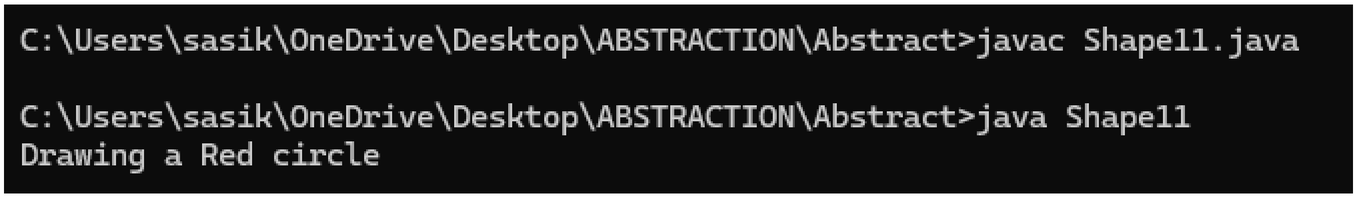
c) Coloring Circle

CODE:

```
abstract class Shape {
    String color;

    Shape(String color) {
        this.color = color;
    }

    abstract void draw();
}

class Circle extends Shape {
    Circle(String color) {
        super(color);
    }

    void draw() {
        System.out.println("Drawing a " + color + " circle");
    }
}
```

```
public class Shape11{
    public static void main(String[] args) {
        Shape shape = new Circle("Red");
        shape.draw();
    }
}
```

SCREENSHOT:



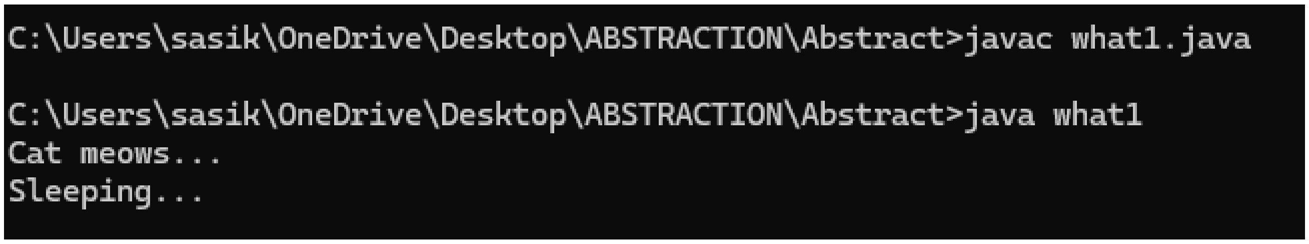
```
C:\Users\sasik\OneDrive\Desktop\ABSTRACTION\Abstract>javac Shape11.java

C:\Users\sasik\OneDrive\Desktop\ABSTRACTION\Abstract>java Shape11
Drawing a Red circle
```

d) Animal characteristics

CODE:

```
public class what1 {
    public static void main(String[] args) {
        Animal myCat = new Cat();
        myCat.makeSound();
        myCat.sleep();
    }
}
abstract class Animal {
    abstract void makeSound();

    void sleep() {
        System.out.println("Sleeping...");
    }
}

class Cat extends Animal {
    void makeSound() {
        System.out.println("Cat meows...");
    }
}
```

SCREENSHOT:



```
C:\Users\sasik\OneDrive\Desktop\ABSTRACTION\Abstract>javac what1.java

C:\Users\sasik\OneDrive\Desktop\ABSTRACTION\Abstract>java what1
Cat meows...
Sleeping...
```

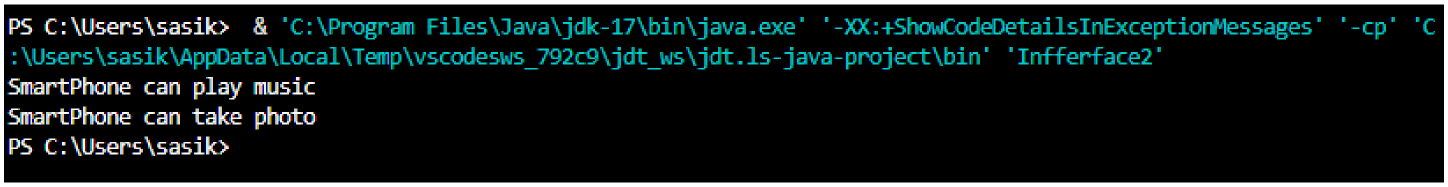
13.INTERFACE

a)Smartphone characteristics

CODE:

```
interface Camara{
    void takePhoto();
}
interface MusicPlayer{
    void canPlayMusic();
}
class SmartPhone implements Camara,MusicPlayer{
    public void takePhoto(){
        System.out.println("SmartPhone can take photo");
    }
    public void canPlayMusic(){
        System.out.println("SmartPhone can play music");
    }
}
public class Infferface2 {
    public static void main(String[] args) {
        SmartPhone s1=new SmartPhone();
        s1.canPlayMusic();
        s1.takePhoto();
    }
}
```

SCREENSHOT:



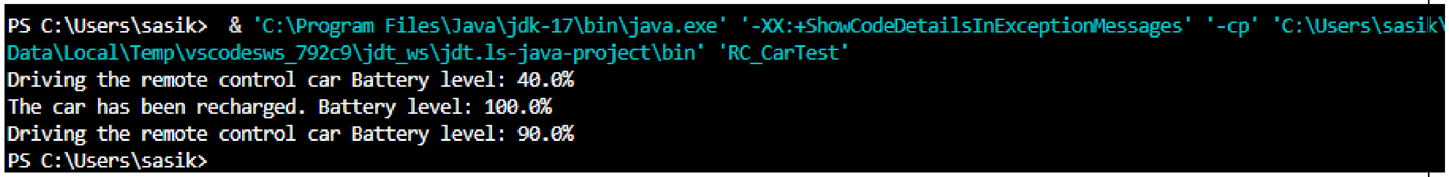
```
PS C:\Users\sasik>  & 'C:\Program Files\Java\jdk-17\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C
:\Users\sasik\AppData\Local\Temp\vscodesws_792c9\jdt_ws\jdt.ls-java-project\bin' 'Infferface2'
SmartPhone can play music
SmartPhone can take photo
PS C:\Users\sasik>
```

b) Testing a Remote control car

CODE:

```
interface RC_Car {
    void drive();
    void recharge();
}

class RemoteControlCar implements RC_Car {
    private double batteryLevel;

    public RemoteControlCar(double batteryLevel) {
        this.batteryLevel = batteryLevel;
    }

    public void drive() {
        if (batteryLevel > 0) {
            batteryLevel -= 10;
            System.out.println("Driving the remote control car Battery
level: " + batteryLevel + "%");
        } else {
            System.out.println("Battery is empty! Please recharge the
car.");
        }
    }

    public void recharge() {
        batteryLevel = 100;
        System.out.println("The car has been recharged. Battery level: "
+ batteryLevel + "%");
    }
}

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

        RemoteControlCar rcCar = new RemoteControlCar(50);


        rcCar.drive();
        rcCar.recharge();
        rcCar.drive();
    }
}
```

SCREENSHOT:



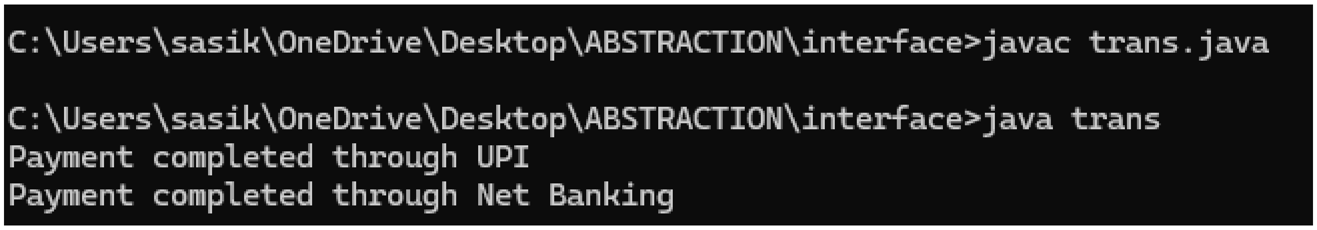
```
PS C:\Users\sasik>  & 'C:\Program Files\Java\jdk-17\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\sasik\
Data\Local\Temp\vscodesws_792c9\jdt_ws\jdt.ls-java-project\bin' 'RC_CarTest'
Driving the remote control car Battery level: 40.0%
The car has been recharged. Battery level: 100.0%
Driving the remote control car Battery level: 90.0%
PS C:\Users\sasik>
```

c) Payment method

CODE:

```
interface Transaction {
    void process();
}

class UPI implements Transaction {
    public void process() {
        System.out.println("Payment completed through UPI");
    }
}

class NetBanking implements Transaction {
    public void process() {
        System.out.println("Payment completed through Net Banking");
    }
}

public class trans {
    public static void main(String[] args) {
        Transaction t1 = new UPI();
        t1.process();

        Transaction t2 = new NetBanking();
        t2.process();
    }
}
```

SCREENSHOT:



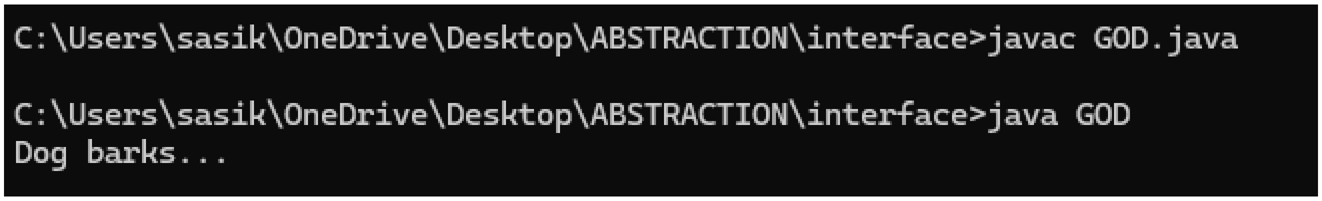
```
C:\Users\sasik\OneDrive\Desktop\ABSTRACTION\interface>javac trans.java

C:\Users\sasik\OneDrive\Desktop\ABSTRACTION\interface>java trans
Payment completed through UPI
Payment completed through Net Banking
```

d) Animal sound

CODE:

```
interface Animal {
    void sound();
}

class Dog implements Animal {
    public void sound() {
        System.out.println("Dog barks...");
    }
}

public class GOD {
    public static void main(String[] args) {
        Animal myDog = new Dog();
        myDog.sound();
    }
}
```

SCREENSHOT:



```
C:\Users\sasik\OneDrive\Desktop\ABSTRACTION\interface>javac GOD.java

C:\Users\sasik\OneDrive\Desktop\ABSTRACTION\interface>java GOD
Dog barks...
```

14.ENCAPSULATION

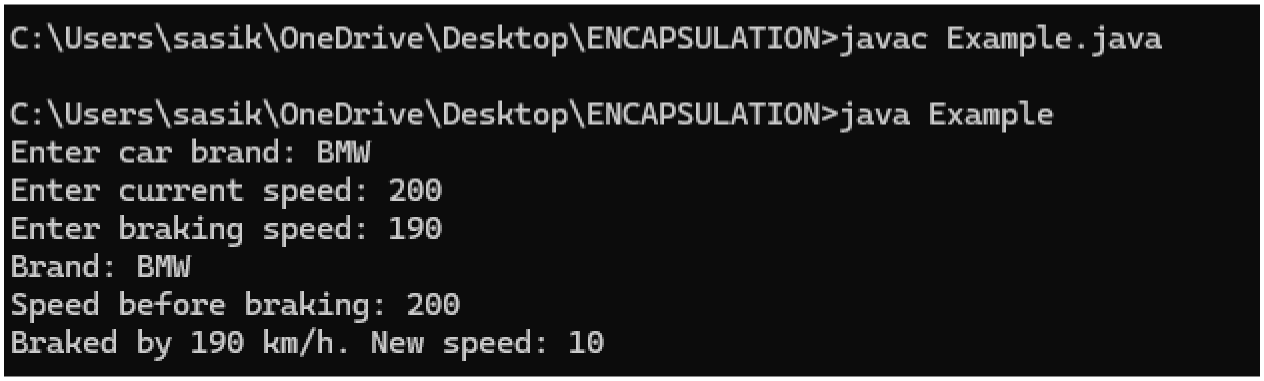
a) Calculating current speed

CODE:

```
import java.util.Scanner;

class Car {
    private String brand;
    private int currentSpeed;

    // Setter for brand
    public void setBrand(String brand) {
        this.brand = brand;
    }

    // Getter for brand
    public String getBrand() {
        return brand;
    }

    // Setter for speed
    public void setSpeed(int speed) {
        if (speed >= 0 && speed <= 200) {
            this.currentSpeed = speed;
        } else {
            System.out.println("Invalid speed! Speed must be between 0
and 200.");
        }
    }

    // Getter for speed
    public int getSpeed() {
        return currentSpeed;
    }

    // Method to slow down the car
    public void brake(int reduceBy) {
        if (reduceBy < 0) {
            System.out.println("Brake value cannot be negative.");
        } else if (currentSpeed - reduceBy >= 0) {
            currentSpeed -= reduceBy;
            System.out.println("Braked by " + reduceBy + " km/h. New
speed: " + currentSpeed);
        } else {
```

```java
                System.out.println("Car is already at minimum speed.");
            }
        }
    }

    public class Example {
        public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);

            Car myCar = new Car();

            System.out.print("Enter car brand: ");
            String brand = sc.nextLine();
            myCar.setBrand(brand);

            System.out.print("Enter current speed: ");
            int speed = sc.nextInt();
            myCar.setSpeed(speed);

            System.out.print("Enter braking speed: ");
            int brake = sc.nextInt();

            System.out.println("Brand: " + myCar.getBrand());
            System.out.println("Speed before braking: " + myCar.getSpeed());
            myCar.brake(brake);

            sc.close();
        }
    }
```

<u>SCREENSHOT:</u>



```
C:\Users\sasik\OneDrive\Desktop\ENCAPSULATION>javac Example.java

C:\Users\sasik\OneDrive\Desktop\ENCAPSULATION>java Example
Enter car brand: BMW
Enter current speed: 200
Enter braking speed: 190
Brand: BMW
Speed before braking: 200
Braked by 190 km/h. New speed: 10
```

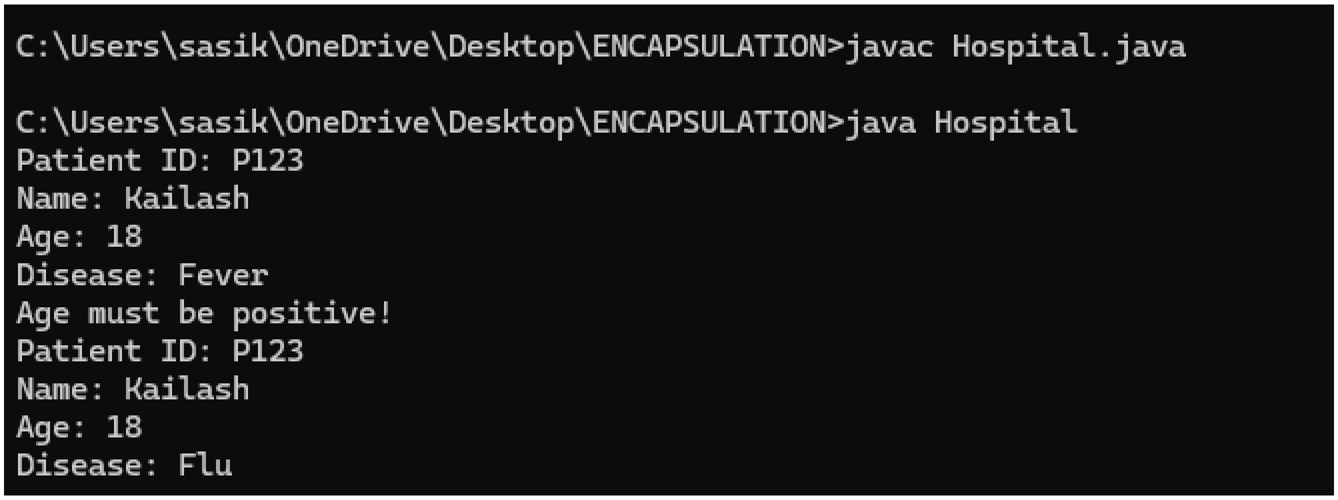
b) Hospital data

CODE:

```
class Patient {
    private String patientID;
    private String name;
    private int age;
    private String disease;

    public Patient(String patientID, String name, int age, String
disease) {
        this.patientID = patientID;
        this.name = name;
        setAge(age);
        this.disease = disease;
    }

    public String getPatientID() {
        return patientID;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    public String getDisease() {
        return disease;
    }

    public void setAge(int age) {
        if (age > 0) {
            this.age = age;
        } else {
            System.out.println("Age must be positive!");
        }
    }

    public void setDisease(String disease) {
        this.disease = disease;
    }
```

```
        public void displayPatientInfo() {
            System.out.println("Patient ID: " + patientID);
            System.out.println("Name: " + name);
            System.out.println("Age: " + age);
            System.out.println("Disease: " + disease);
        }
    }

    public class Hospital{
        public static void main(String[] args) {
            Patient p = new Patient("P123", "Kailash", 18, "Fever");
            p.displayPatientInfo();
            p.setAge(-5);
            p.setDisease("Flu");
            p.displayPatientInfo();
        }
    }
```

SCREENSHOT:



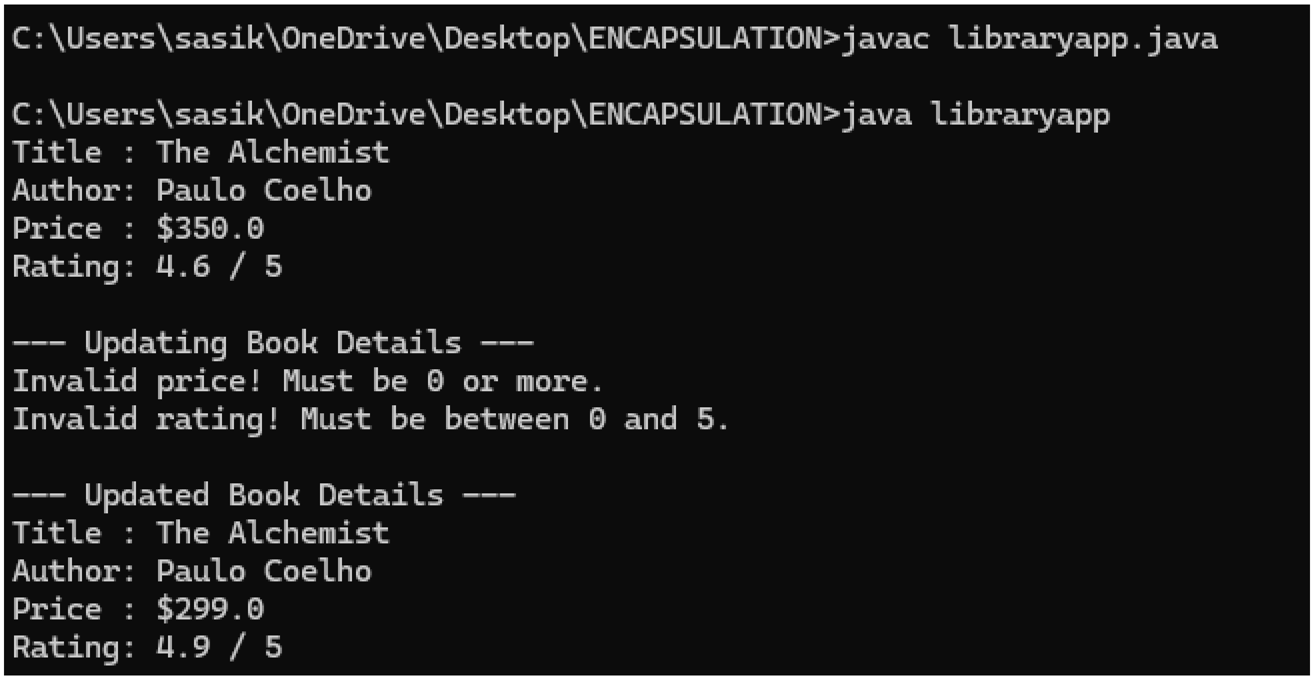
```
C:\Users\sasik\OneDrive\Desktop\ENCAPSULATION>javac Hospital.java

C:\Users\sasik\OneDrive\Desktop\ENCAPSULATION>java Hospital
Patient ID: P123
Name: Kailash
Age: 18
Disease: Fever
Age must be positive!
Patient ID: P123
Name: Kailash
Age: 18
Disease: Flu
```

c) Library App

CODE:

```
class Book {

    private String title;
    private String author;
    private double price;
    private double rating;


    public Book(String title, String author, double price, double
rating) {
        this.title = title;
        this.author = author;
        setPrice(price);
        setRating(rating);
    }

    // Public getters
    public String getTitle() {
        return title;
    }

    public String getAuthor() {
        return author;
    }

    public double getPrice() {
        return price;
    }

    public double getRating() {
        return rating;
    }


    public void setPrice(double price) {
        if (price >= 0) {
            this.price = price;
        } else {
            System.out.println("Invalid price! Must be 0 or more.");
        }
    }
    public void setRating(double rating) {
        if (rating >= 0 && rating <= 5) {
            this.rating = rating;
        } else {
            System.out.println("Invalid rating! Must be between 0 and
5.");
        }
    }

    public void displayBook() {
        System.out.println("Title : " + title);
        System.out.println("Author: " + author);
        System.out.println("Price : $" + price);
        System.out.println("Rating: " + rating + " / 5");
    }
}
public class libraryapp {
    public static void main(String[] args) {
        Book book = new Book("The Alchemist", "Paulo Coelho", 350.0,
4.6);
        book.displayBook();

        System.out.println("\n--- Updating Book Details ---");
        book.setPrice(-100);
        book.setRating(6);
        book.setPrice(299.0);
        book.setRating(4.9);

        System.out.println("\n--- Updated Book Details ---");
        book.displayBook();
    }
}
```

SCREENSHOT:



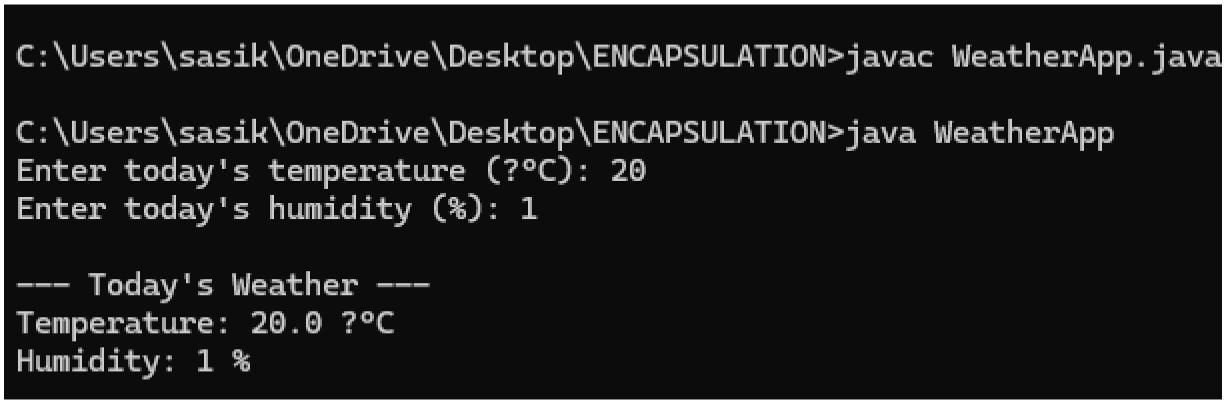
```
C:\Users\sasik\OneDrive\Desktop\ENCAPSULATION>javac libraryapp.java

C:\Users\sasik\OneDrive\Desktop\ENCAPSULATION>java libraryapp
Title : The Alchemist
Author: Paulo Coelho
Price : $350.0
Rating: 4.6 / 5

--- Updating Book Details ---
Invalid price! Must be 0 or more.
Invalid rating! Must be between 0 and 5.

--- Updated Book Details ---
Title : The Alchemist
Author: Paulo Coelho
Price : $299.0
Rating: 4.9 / 5
```

d) Weather App

CODE:

```
import java.util.Scanner;

class Weather {
    private double temperature;
    private int humidity;

    public Weather(double temperature, int humidity) {
        setTemperature(temperature);
        setHumidity(humidity);
    }

    public void setTemperature(double temperature) {
        if (temperature >= -50 && temperature <= 60) {
            this.temperature = temperature;
        } else {
            System.out.println("Invalid temperature! Must be between -50
and 60°C.");
        }
    }

    public void setHumidity(int humidity) {
        if (humidity >= 0 && humidity <= 100) {
            this.humidity = humidity;
        } else {
            System.out.println("Invalid humidity! Must be between 0% and 
100%.");
        }
    }

    public double getTemperature() {
        return temperature;
    }

    public int getHumidity() {
        return humidity;
    }

    public void displayWeather() {
        System.out.println("Temperature: " + temperature + " °C");
        System.out.println("Humidity: " + humidity + " %");
    }
}
public class WeatherApp {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter today's temperature (°C): ");
        double temp = sc.nextDouble();

        System.out.print("Enter today's humidity (%): ");
        int hum = sc.nextInt();

        Weather today = new Weather(temp, hum);

        System.out.println("\n--- Today's Weather ---");
        today.displayWeather();

        sc.close();
    }
}
```

SCREENSHOT:



```
C:\Users\sasik\OneDrive\Desktop\ENCAPSULATION>javac WeatherApp.java

C:\Users\sasik\OneDrive\Desktop\ENCAPSULATION>java WeatherApp
Enter today's temperature (?°C): 20
Enter today's humidity (%): 1

--- Today's Weather ---
Temperature: 20.0 ?°C
Humidity: 1 %
```

15. PACKAGES PROGRAMS

a) Online food delivery

CODE:

1.Main

```
public class Main {
    public static void main(String[] args) {
        // Create menu items
        MenuItem pizza = new MenuItem("Pizza", 8.99);
        MenuItem burger = new MenuItem("Burger", 5.49);
        MenuItem fries = new MenuItem("Fries", 2.99);

        // Display menu
        System.out.println("Menu:");
        pizza.displayItem();
        burger.displayItem();
        fries.displayItem();

        // Create an order
        Order myOrder = new Order();
        myOrder.addItem(pizza);
        myOrder.addItem(fries);

        // Show order summary
        myOrder.displayOrder();

        // Delivery
        Delivery myDelivery = new Delivery("123 Main Street");
        myDelivery.showStatus();

        // Simulate delivery update
        myDelivery.updateStatus("On the way");
        myDelivery.showStatus();

        myDelivery.updateStatus("Delivered");
        myDelivery.showStatus();
    }
}
```

2.Menu

```
public class MenuItem {
    private String name;
    private double price;

    public MenuItem(String name, double price) {
        this.name = name;
        this.price = price;
    }

    public String getName() {
        return name;
    }

    public double getPrice() {
        return price;
    }

    public void displayItem() {
        System.out.println(name + " - $" + price);
    }
}
```

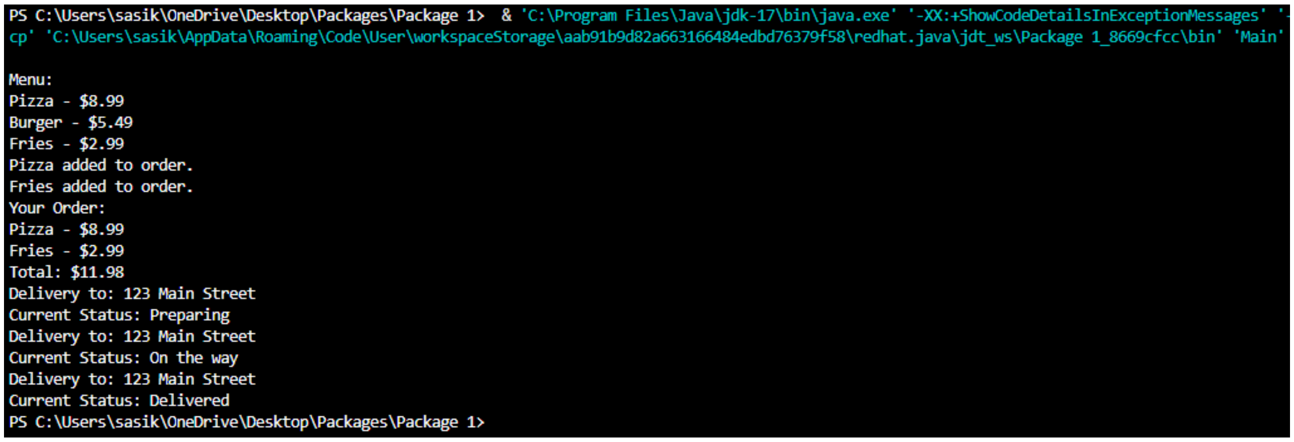
3.Ordering

```
import java.util.ArrayList;
import java.util.List;

public class Order {
    private List<MenuItem> items;

    public Order() {
        items = new ArrayList<>();
    }

    public void addItem(MenuItem item) {
        items.add(item);
        System.out.println(item.getName() + " added to order.");
    }

    public double calculateTotal() {
        double total = 0;
        for (MenuItem item : items) {
            total += item.getPrice();
        }
        return total;
    }

    public void displayOrder() {
        System.out.println("Your Order:");
        for (MenuItem item : items) {
            item.displayItem();
        }
        System.out.println("Total: $" + calculateTotal());
    }
}
```

4.Delivery

```
public class Delivery {
    private String address;
    private String status;

    public Delivery(String address) {
        this.address = address;
        this.status = "Preparing";
    }

    public void updateStatus(String newStatus) {
        status = newStatus;
    }

    public void showStatus() {
        System.out.println("Delivery to: " + address);
        System.out.println("Current Status: " + status);
    }
}
```

SCREENSHOT:



```
PS C:\Users\sasik\OneDrive\Desktop\Packages\Package 1>  & 'C:\Program Files\Java\jdk-17\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-
cp' 'C:\Users\sasik\AppData\Roaming\Code\User\workspaceStorage\aab91b9d82a663166484edbd76379f58\redhat.java\jdt_ws\Package 1_8669cfcc\bin' 'Main'

Menu:
Pizza - $8.99
Burger - $5.49
Fries - $2.99
Pizza added to order.
Fries added to order.
Your Order:
Pizza - $8.99
Fries - $2.99
Total: $11.98
Delivery to: 123 Main Street
Current Status: Preparing
Delivery to: 123 Main Street
Current Status: On the way
Delivery to: 123 Main Street
Current Status: Delivered
PS C:\Users\sasik\OneDrive\Desktop\Packages\Package 1>
```

b) Calculator app

CODE:

1.Main

```
package calculator;
import simple.SimpleCalculator;
import scientific.ScientificCalculator;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        SimpleCalculator basic = new SimpleCalculator();
        ScientificCalculator sci = new ScientificCalculator();
        Scanner sc = new Scanner(System.in);

        System.out.println("Welcome to Multi-Package Super
Calculator!");

        while (true) {
            System.out.println("\nChoose operation:");
            System.out.println(" +  -
  *  /  %  sqrt  ^  sin  cos  tan  log  ln ");
            System.out.println("Or type 'exit' to quit.");
            String op = sc.next();

            if (op.equalsIgnoreCase("exit")) {
                System.out.println("Goodbye!");
                break;
            }
            try {
                double num1, num2, result = 0;
                boolean valid = true;

                if (op.equals("sqrt") || op.equals("sin") ||
op.equals("cos") || op.equals("tan") || op.equals("log") ||
op.equals("ln")) {
                    System.out.println("Enter number:");
                    num1 = getValidDouble(sc);

                    switch (op) {
                        case "sqrt":
                            result = sci.sqrt(num1);
                            break;
                        case "sin":
                            result = sci.sin(num1);
                            break;
                        case "cos":
                            result = sci.cos(num1);
                            break;
                        case "tan":
                            result = sci.tan(num1);
                            break;
                        case "log":
                            result = sci.log(num1);
                            break;
                        case "ln":
                            result = sci.ln(num1);
                            break;
                        default:
                            valid = false;
                    }
                } else {
                    System.out.println("Enter first number:");
                    num1 = getValidDouble(sc);
                    System.out.println("Enter second number:");
                    num2 = getValidDouble(sc);

                    switch (op) {
                        case "+":
                            result = basic.add(num1, num2);
                            break;
                        case "-":
                            result = basic.subtract(num1, num2);
                            break;
                        case "*":
                            result = basic.multiply(num1, num2);
                            break;
                        case "/":
                            result = basic.divide(num1, num2);
                            break;
                        case "%":
                            result = basic.modulus(num1, num2);
                            break;
                        case "^":
                            result = sci.power(num1, num2);
                            break;
                        default:
                            valid = false;
                    }
                }

                if (valid) {
                    System.out.println("Result: " + result);
                } else {
                    System.out.println("Invalid operation.");
                }
            } catch (ArithmeticException e) {
                System.out.println("Error: " + e.getMessage());
            }
        }
        sc.close();
    }

    private static double getValidDouble(Scanner sc) {
        while (!sc.hasNextDouble()) {
            System.out.println("Invalid input. Enter a number:");
            sc.next();
        }
        return sc.nextDouble();
    }
}
```

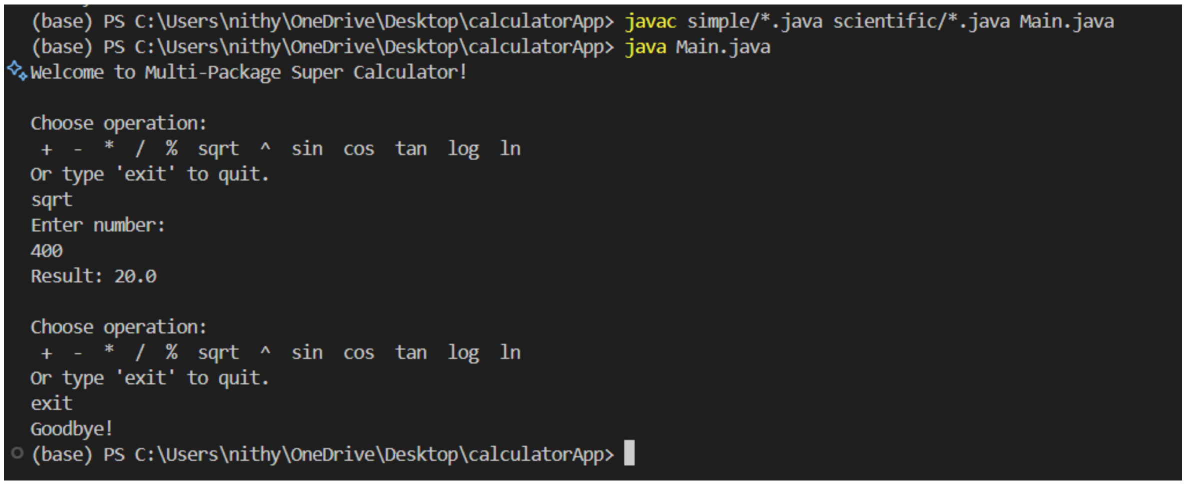
2.Simple calculator

```
package simple;

public class SimpleCalculator {
    public double add(double a, double b) {
        return a + b;
    }

    public double subtract(double a, double b) {
        return a - b;
    }

    public double multiply(double a, double b) {
        return a * b;
    }

    public double divide(double a, double b) {
        if (b == 0) {
            throw new ArithmeticException("Division by zero is not
allowed.");
        }
        return a / b;
    }

    public double modulus(double a, double b) {
        return a % b;
    }
}
```

3.Scientific calculator

```
package scientific;

public class ScientificCalculator {
    public double sqrt(double a) {
        return Math.sqrt(a);
    }

    public double power(double a, double b) {
        return Math.pow(a, b);
    }

    public double sin(double degree) {
        return Math.sin(Math.toRadians(degree));
    }

    public double cos(double degree) {
        return Math.cos(Math.toRadians(degree));
    }

    public double tan(double degree) {
        return Math.tan(Math.toRadians(degree));
    }

    public double log(double a) {
        return Math.log10(a);
    }

    public double ln(double a) {
        return Math.log(a);
    }
}
```

SCREENSHOT:



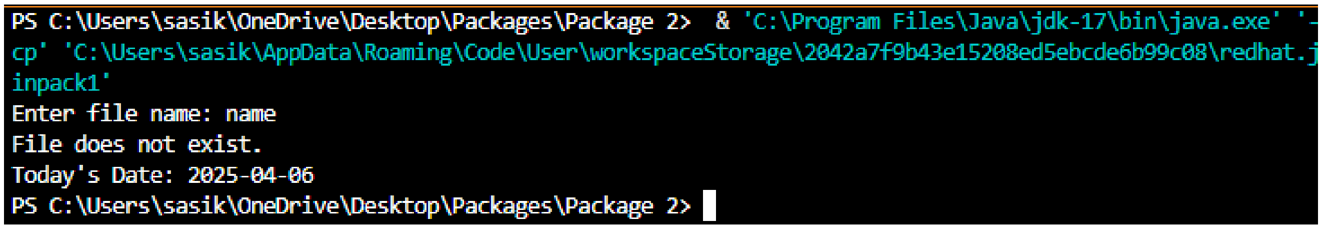
```
(base) PS C:\Users\nithy\OneDrive\Desktop\calculatorApp> javac simple/*.java scientific/*.java Main.java
(base) PS C:\Users\nithy\OneDrive\Desktop\calculatorApp> java Main.java
Welcome to Multi-Package Super Calculator!

Choose operation:
 +  -  *  /  %  sqrt  ^  sin  cos  tan  log  ln
Or type 'exit' to quit.
sqrt
Enter number:
400
Result: 20.0

Choose operation:
 +  -  *  /  %  sqrt  ^  sin  cos  tan  log  ln
Or type 'exit' to quit.
exit
Goodbye!
(base) PS C:\Users\nithy\OneDrive\Desktop\calculatorApp>
```

c)Local date

CODE:

```
import java.util.Scanner;
import java.io.File;
import java.time.LocalDate;

class builtinpack1 {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter file name: ");
        String fileName = sc.nextLine();


        File file = new File(fileName);
        if (file.exists()) {
            System.out.println("File exists: " + file.getName());
        } else {
            System.out.println("File does not exist.");
        }


        LocalDate today = LocalDate.now();
        System.out.println("Today's Date: " + today);

        sc.close();
    }
}
```

SCREENSHOTS:



```
PS C:\Users\sasik\OneDrive\Desktop\Packages\Package 2>  & 'C:\Program Files\Java\jdk-17\bin\java.exe' '-
cp' 'C:\Users\sasik\AppData\Roaming\Code\User\workspaceStorage\2042a7f9b43e15208ed5ebcde6b99c08\redhat.j
inpack1'
Enter file name: name
File does not exist.
Today's Date: 2025-04-06
PS C:\Users\sasik\OneDrive\Desktop\Packages\Package 2>
```

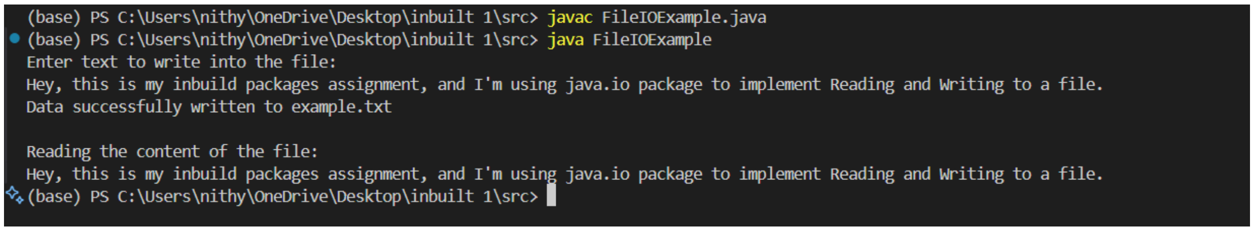
d)Reading and Writing a file

CODE:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class FileIOExample {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String filename = "example.txt";

        try {
            System.out.println("Enter text to write into the file:");
            String userInput = scanner.nextLine();

            FileWriter writer = new FileWriter(filename);
            writer.write(userInput);
            writer.close();
```

```
                System.out.println("Data successfully written to " +
filename);

                System.out.println("\nReading the content of the file:");
                BufferedReader reader = new BufferedReader(new
FileReader(filename));
                String line;
                while ((line = reader.readLine()) != null) {
                    System.out.println(line);
                }
                reader.close();

            } catch (IOException e) {
                System.out.println("An error occurred: " +
e.getMessage());
            } finally {
                scanner.close();
            }
        }
}
```

SCREENSHOT:



```
(base) PS C:\Users\nithy\OneDrive\Desktop\inbuilt 1\src> javac FileIOExample.java
(base) PS C:\Users\nithy\OneDrive\Desktop\inbuilt 1\src> java FileIOExample
Enter text to write into the file:
Hey, this is my inbuild packages assignment, and I'm using java.io package to implement Reading and Writing to a file.
Data successfully written to example.txt

Reading the content of the file:
Hey, this is my inbuild packages assignment, and I'm using java.io package to implement Reading and Writing to a file.
(base) PS C:\Users\nithy\OneDrive\Desktop\inbuilt 1\src>
```

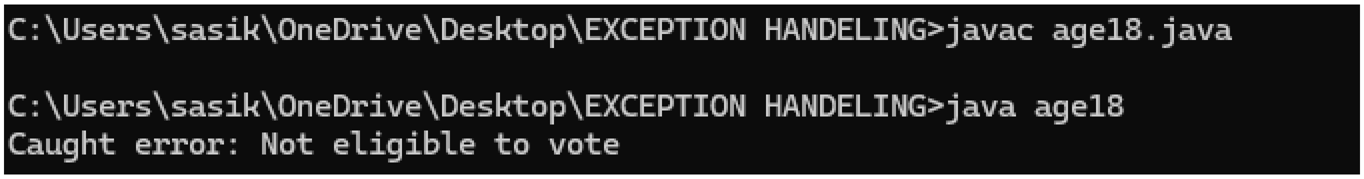
16. EXCEPTION HANDLING

a)Voting Eligibility

CODE:

```
public class age18 {
    public static void main(String[] args) {
        try {
            checkAge(11);
        } catch (IllegalArgumentException e) {
            System.out.println("Caught error: " + e.getMessage());
        }
    }

    public static void checkAge(int age) {
        if (age < 18) {
            throw new IllegalArgumentException("Not eligible to vote");
        }
        System.out.println("Access granted. Eligible to vote.");
    }
}
```

SCREENSHOTS:



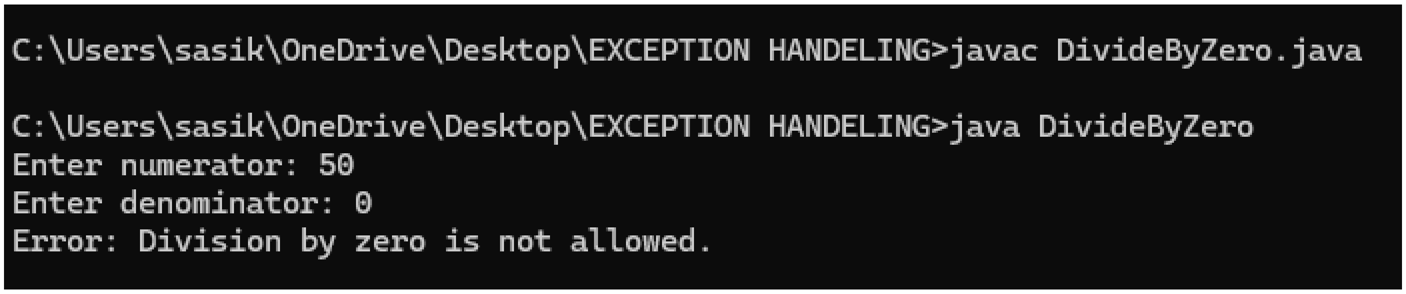
```
C:\Users\sasik\OneDrive\Desktop\EXCEPTION HANDELING>javac age18.java

C:\Users\sasik\OneDrive\Desktop\EXCEPTION HANDELING>java age18
Caught error: Not eligible to vote
```

b) Divide by Zero error

CODE:

```
import java.util.Scanner;

public class DivideByZero {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Enter numerator: ");
            int num = scanner.nextInt();
            System.out.print("Enter denominator: ");
            int den = scanner.nextInt();

            int result = num / den;
            System.out.println("Result: " + result);

        } catch (ArithmeticException e) {
            System.out.println("Error: Division by zero is not allowed.");
        }

        scanner.close();
    }
}
```

<u>SCREENSHOTS</u>



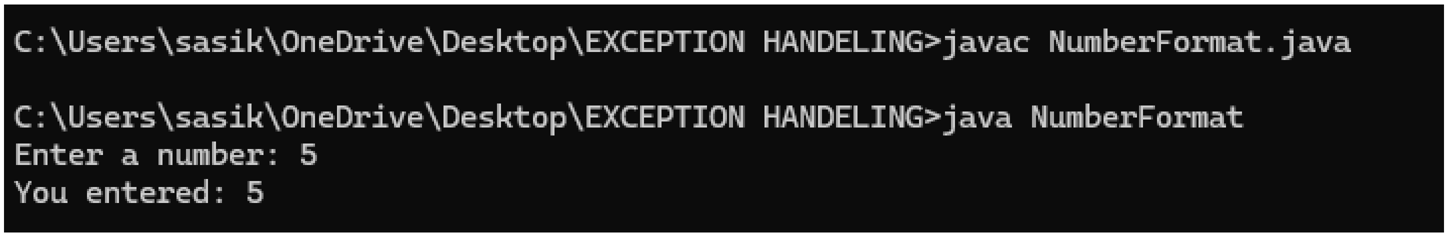
```
C:\Users\sasik\OneDrive\Desktop\EXCEPTION HANDELING>javac DivideByZero.java

C:\Users\sasik\OneDrive\Desktop\EXCEPTION HANDELING>java DivideByZero
Enter numerator: 50
Enter denominator: 0
Error: Division by zero is not allowed.
```

c)Number format

<u>CODE:</u>

```
import java.util.Scanner;

public class NumberFormat {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Enter a number: ");
            String input = scanner.nextLine();
            int num = Integer.parseInt(input);
            System.out.println("You entered: " + num);

        } catch (NumberFormatException e) {
            System.out.println("Error: Invalid number format!");
        }
        scanner.close();
    }
}
```

SCREENSHOTS



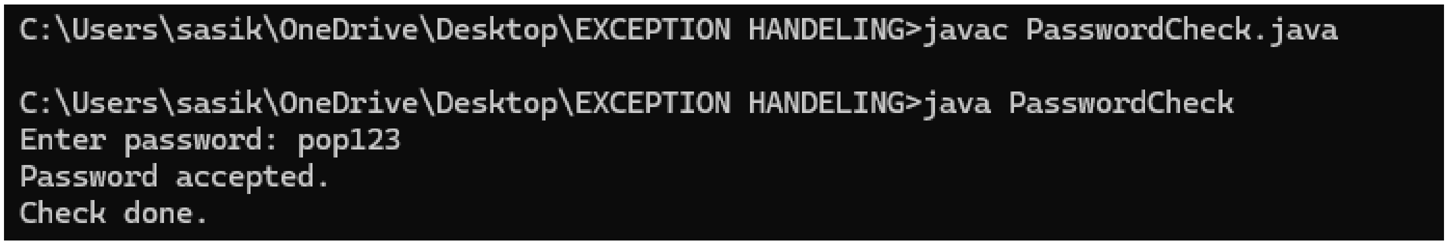
```
C:\Users\sasik\OneDrive\Desktop\EXCEPTION HANDELING>javac NumberFormat.java

C:\Users\sasik\OneDrive\Desktop\EXCEPTION HANDELING>java NumberFormat
Enter a number: 5
You entered: 5
```

d)Check Password

CODE:

```
import java.util.Scanner;

class WeakPasswordException extends Exception {
    public WeakPasswordException(String message) {
        super(message);
    }
}

public class PasswordCheck {

    public static void check(String password) throws WeakPasswordException
{
        if (password.length() < 6) {
            throw new WeakPasswordException("Password too short! Must be at
least 6 characters.");
        }
        System.out.println("Password accepted.");
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        try {
            System.out.print("Enter password: ");
            String pass = sc.nextLine();
            check(pass);
        } catch (WeakPasswordException e) {
            System.out.println("Error: " + e.getMessage());
        } finally {
            System.out.println("Check done.");
            sc.close();
        }
    }
}
```

SCREENSHOTS:



```
C:\Users\sasik\OneDrive\Desktop\EXCEPTION HANDELING>javac PasswordCheck.java

C:\Users\sasik\OneDrive\Desktop\EXCEPTION HANDELING>java PasswordCheck
Enter password: pop123
Password accepted.
Check done.
```

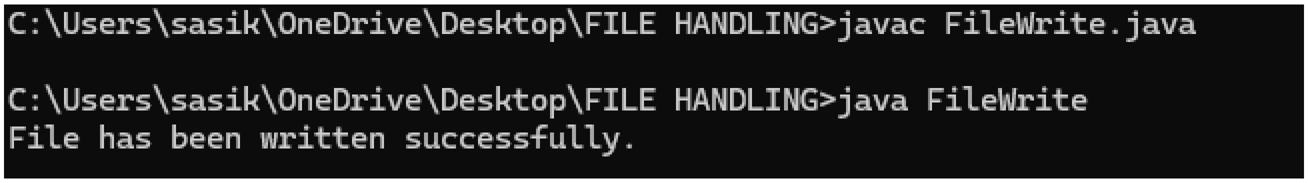
17. FILE HANDLING PROGRAMS

a)Write into File

CODE:

```
import java.io.FileWriter;
import java.io.IOException;

public class FileWrite {
    public static void main(String[] args) {
        String message = "Hello, My name is sussy";

        try {
            FileWriter fw = new FileWriter("example.txt");
            fw.write(message);
            fw.close();
            System.out.println("File has been written successfully.");
        } catch (IOException ex) {
            System.out.println("Something went wrong while writing the
file.");
        }
    }
}
```

SCREENSHOTS:



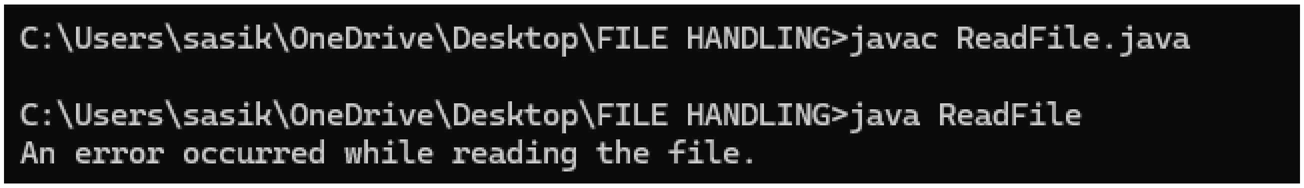
```
C:\Users\sasik\OneDrive\Desktop\FILE HANDLING>javac FileWrite.java

C:\Users\sasik\OneDrive\Desktop\FILE HANDLING>java FileWrite
File has been written successfully.
```

b)Read into File

CODE:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class ReadFile {
    public static void main(String[] args) {
        String path = "\"C:\\Users\\sasik\\OneDrive\\Documents\\New Text
Document.txt\"";

        try {
            FileReader r = new FileReader(path);
            BufferedReader reader = new BufferedReader(r);
            String line;

            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }

            reader.close();
        } catch (IOException ex) {
            System.out.println("An error occurred while reading the
file.");
        }
    }
}
```

SCREENSHOTS:



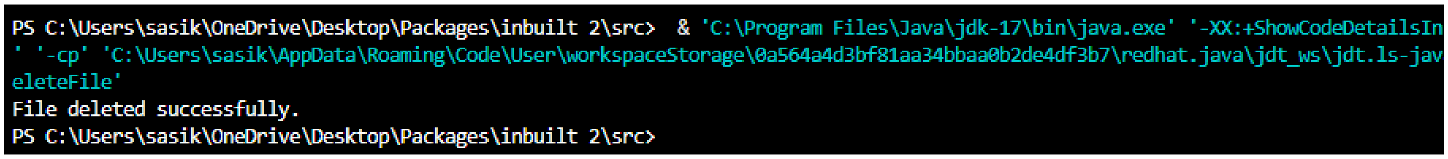
```
C:\Users\sasik\OneDrive\Desktop\FILE HANDLING>javac ReadFile.java

C:\Users\sasik\OneDrive\Desktop\FILE HANDLING>java ReadFile
An error occurred while reading the file.
```

c)Deleting File

CODE:

```
import java.io.File;

public class DeleteFile {
    public static void main(String[] args) {
        File file = new File("example.txt");
        if (file.delete()) {
            System.out.println("File deleted successfully.");
        } else {
            System.out.println("Failed to delete the file.");
        }
    }
}
```

SCREENSHOTS:



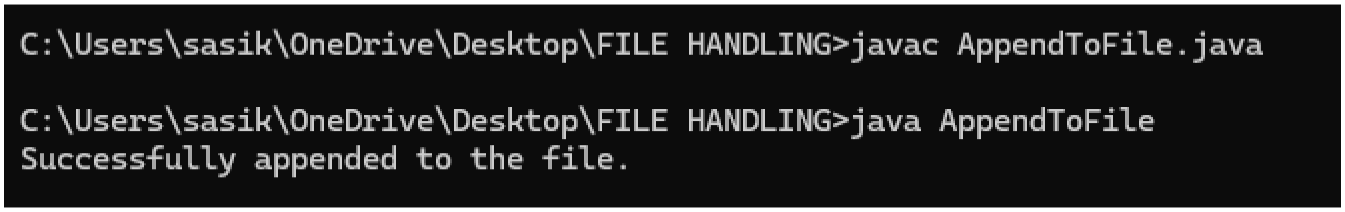
```
PS C:\Users\sasik\OneDrive\Desktop\Packages\inbuilt 2\src>  & 'C:\Program Files\Java\jdk-17\bin\java.exe' '-XX:+ShowCodeDetailsIn
' '-cp' 'C:\Users\sasik\AppData\Roaming\Code\User\workspaceStorage\0a564a4d3bf81aa34bbaa0b2de4df3b7\redhat.java\jdt_ws\jdt.ls-jav
eleteFile'
File deleted successfully.
PS C:\Users\sasik\OneDrive\Desktop\Packages\inbuilt 2\src>
```

d)Appending File

CODE:

```
import java.io.FileWriter;
import java.io.IOException;

public class AppendToFile {
    public static void main(String[] args) {
        try {
            FileWriter writer = new FileWriter("example.txt", true);
            writer.write("\nI'm from Chennai");
            writer.close();
            System.out.println("Successfully appended to the file.");
        } catch (IOException e) {
            System.out.println("An error occurred.");
            e.printStackTrace();
        }
    }
}
```

SCREENSHOTS:



```
C:\Users\sasik\OneDrive\Desktop\FILE HANDLING>javac AppendToFile.java

C:\Users\sasik\OneDrive\Desktop\FILE HANDLING>java AppendToFile
Successfully appended to the file.
```