LAB RECORD

23CSE111- Object Oriented Programming

Submitted by

CH.SC.U4CSE24128 -M. charitha varshini

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING

CHENNAI

March - 2025

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.U4CSE24128 – M. charitha varshini*** 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 / /2025

Internal Examiner 1 Internal Examiner 2

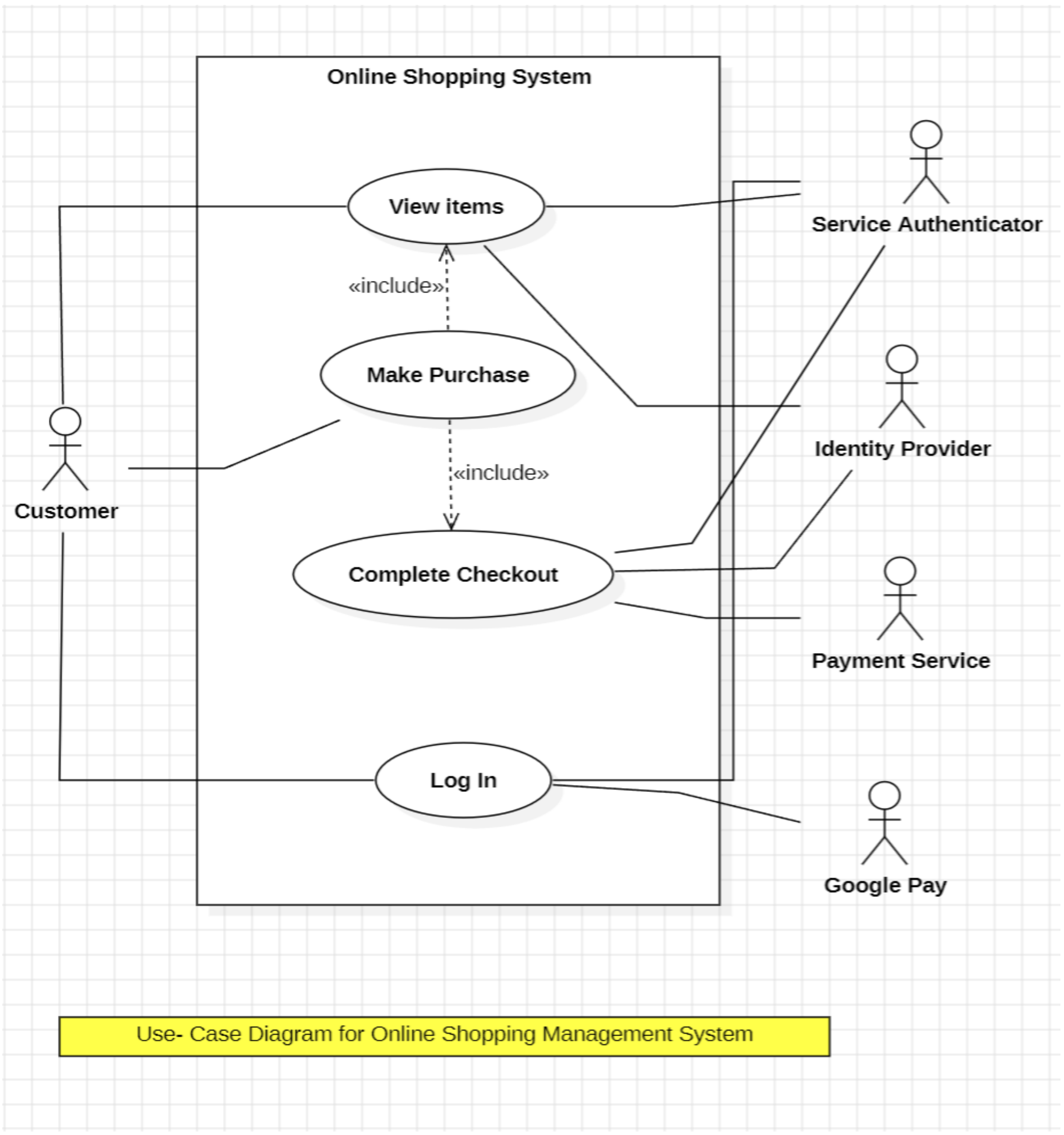
M. Charitha Varshini
CH.SC.U4CSE24128

INDEX

UML DIAGRAMS

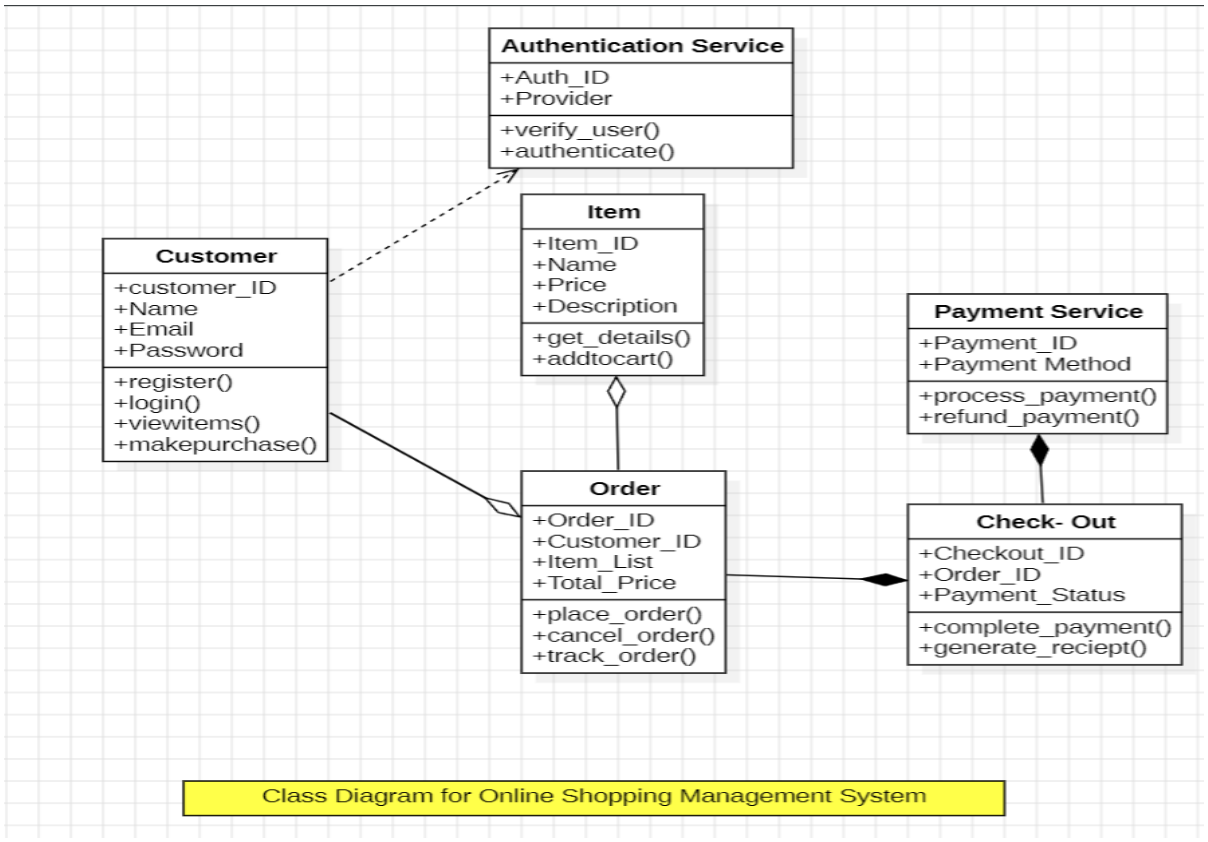
1. ONLINE SHOPPING MANAGEMENT SYSTEM

1. a) Use Case Diagram:



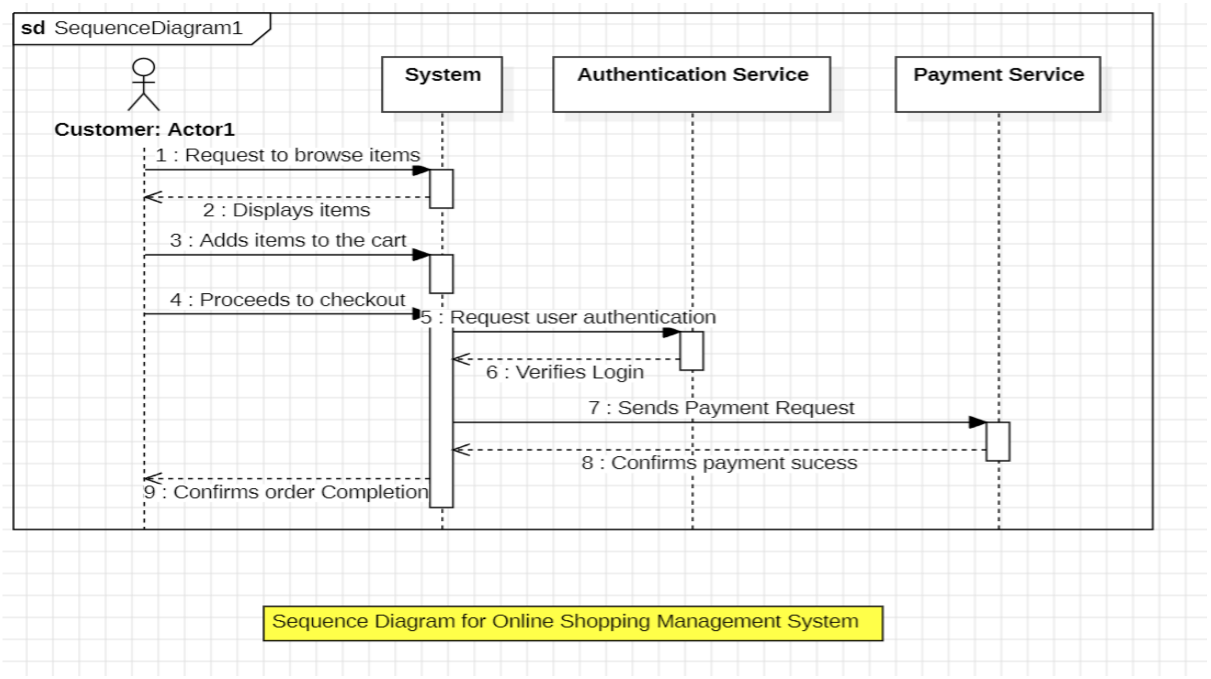
Use- Case Diagram for Online Shopping Management System

1. b) Class Diagram:



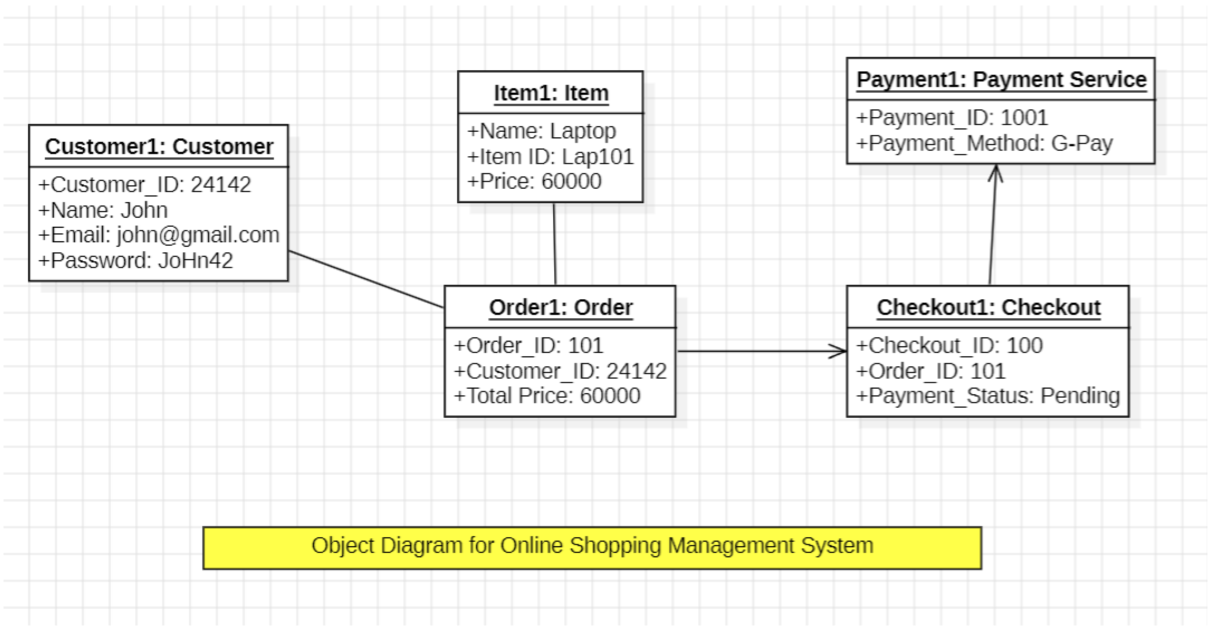
Class Diagram for Online Shopping Management System

1. c) Sequence Diagram:

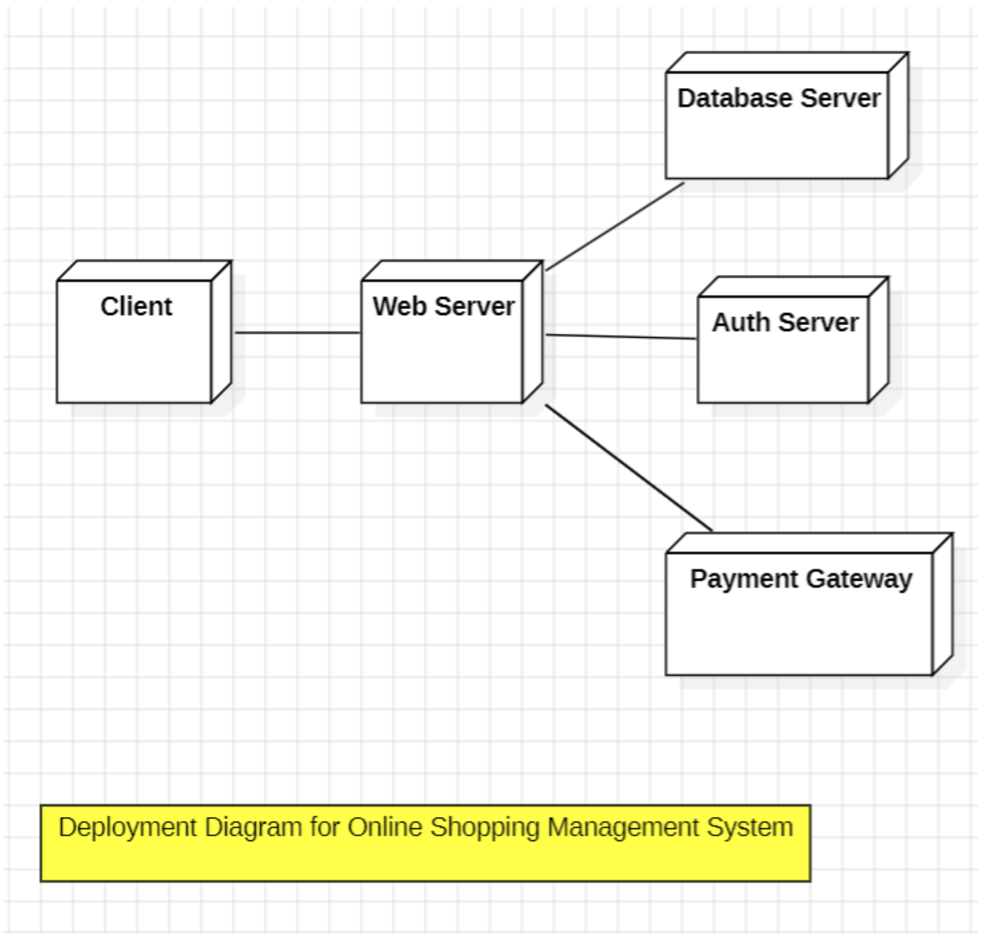

sd SequenceDiagram1
Customer: Actor1
System
Authentication Service
Payment Service
1 : Request to browse items
2 : Displays items
3 : Adds items to the cart
4 : Proceeds to checkout
5 : Request user authentication
6 : Verifies Login
7 : Sends Payment Request
8 : Confirms payment sucess
9 : Confirms order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

Sequence Diagram for Online Shopping Management System

1. d) Object Diagram:

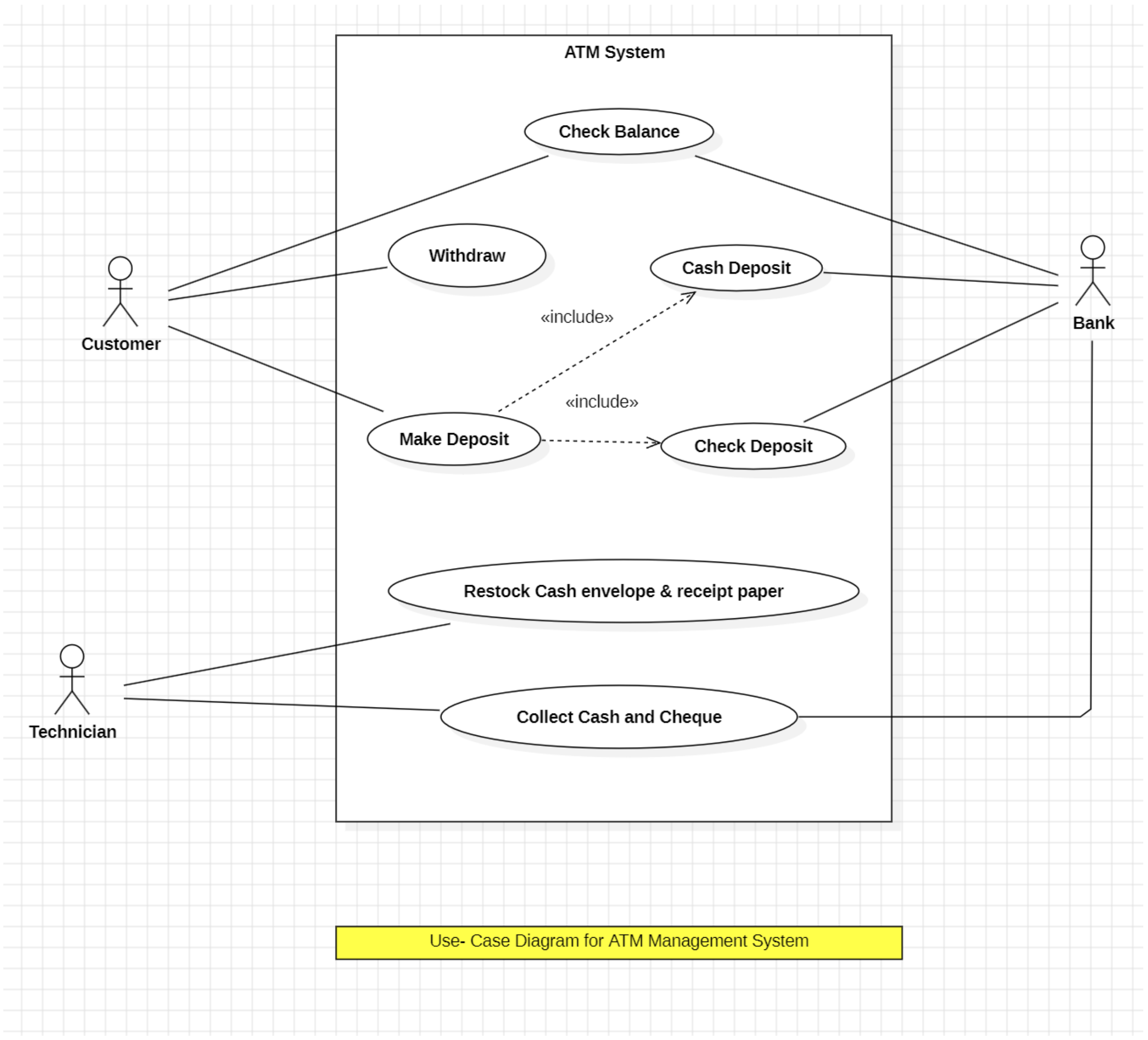


1. e) Deployment Diagram:



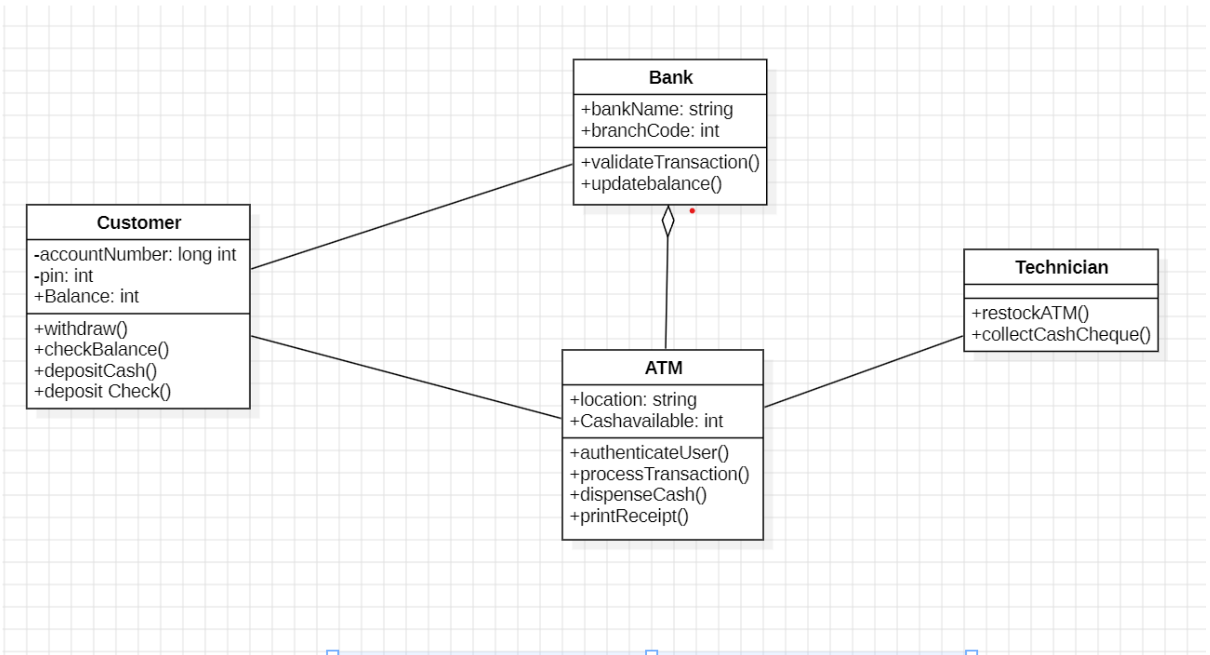
2. ATM MANAGEMENT SYSTEM

2.a) Use Case Diagram:



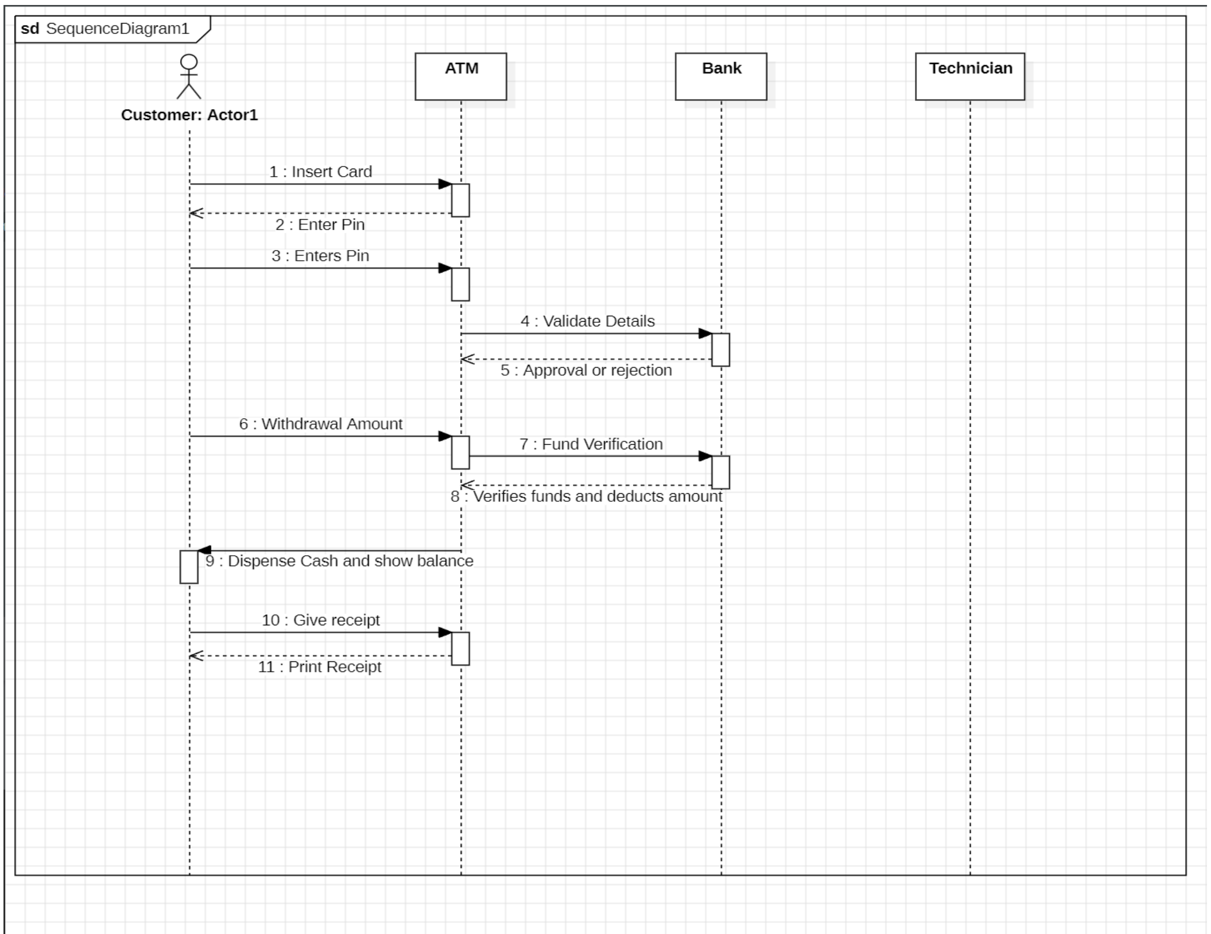
Use- Case Diagram for ATM Management System

2.b) Class Diagram:



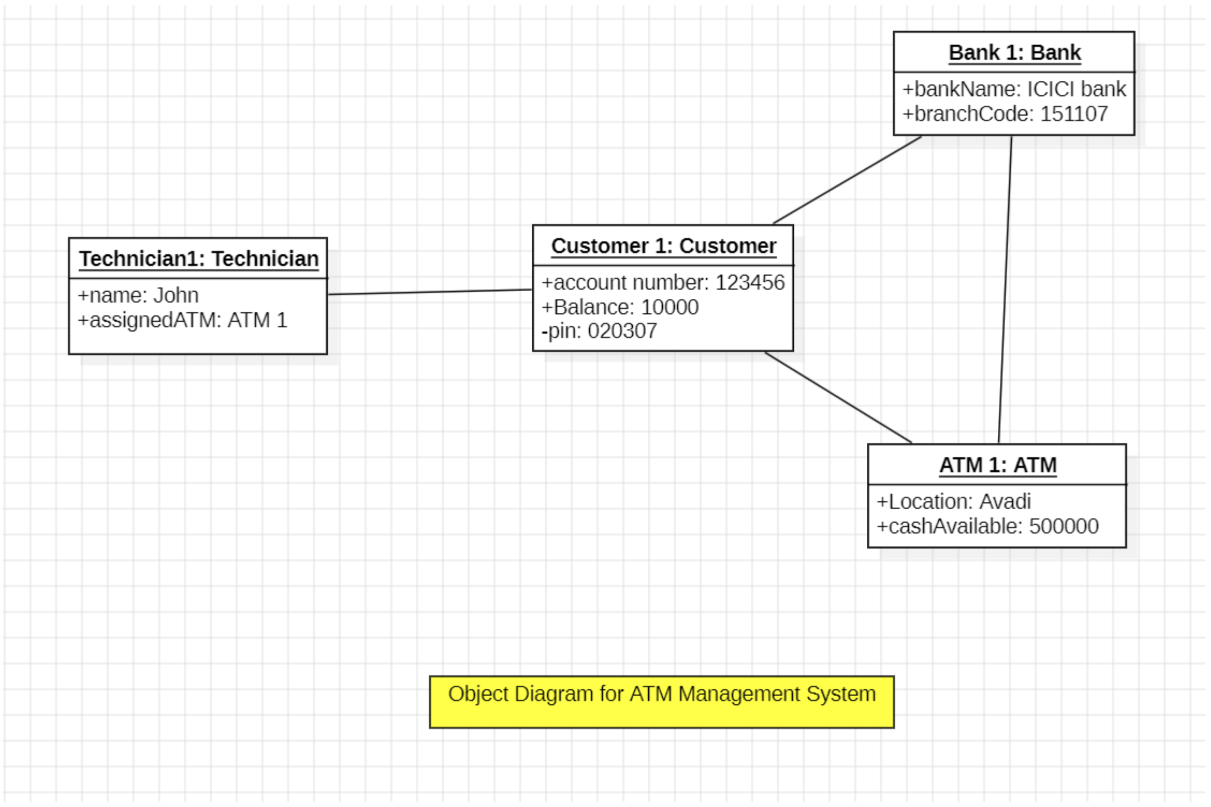
Class Diagram for ATM Management System

2. c) Sequence Diagram:

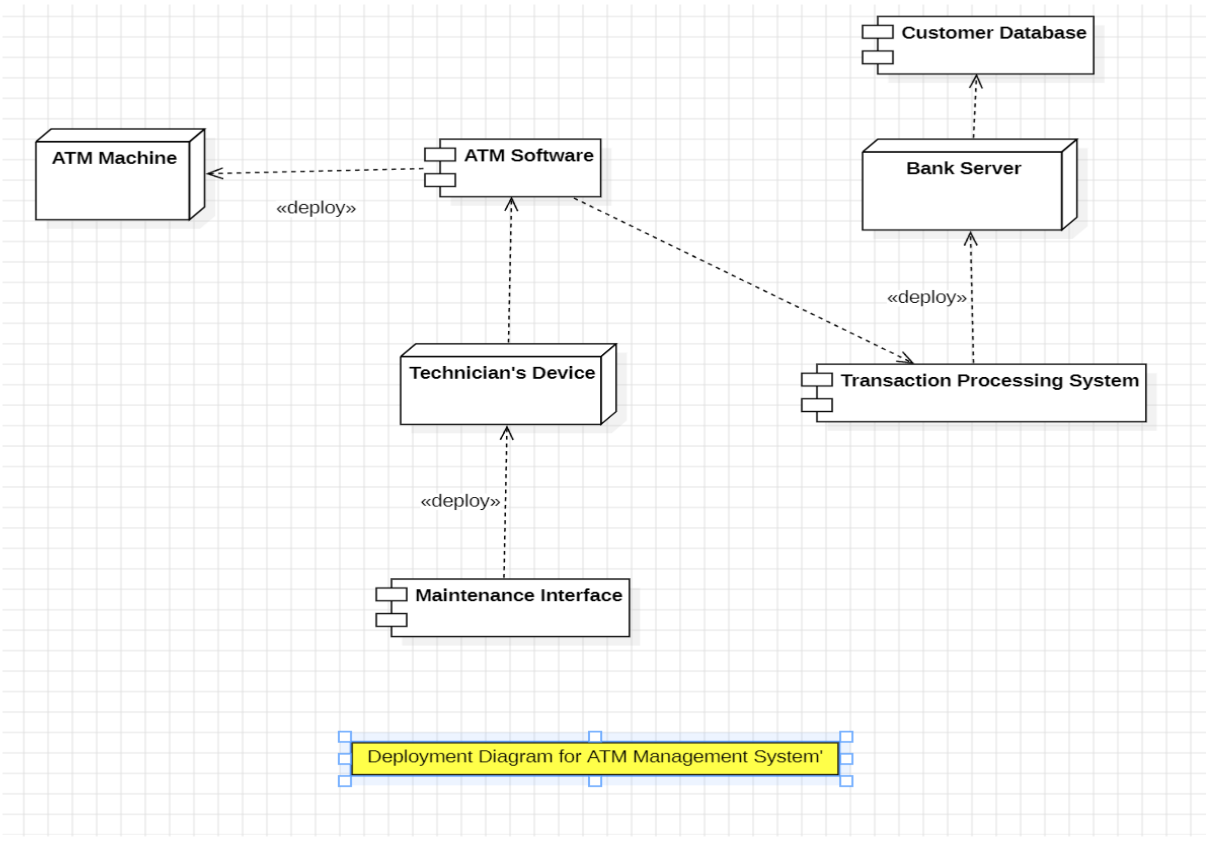


Sequence Diagram for ATM Management System

2.d) Object Diagram:



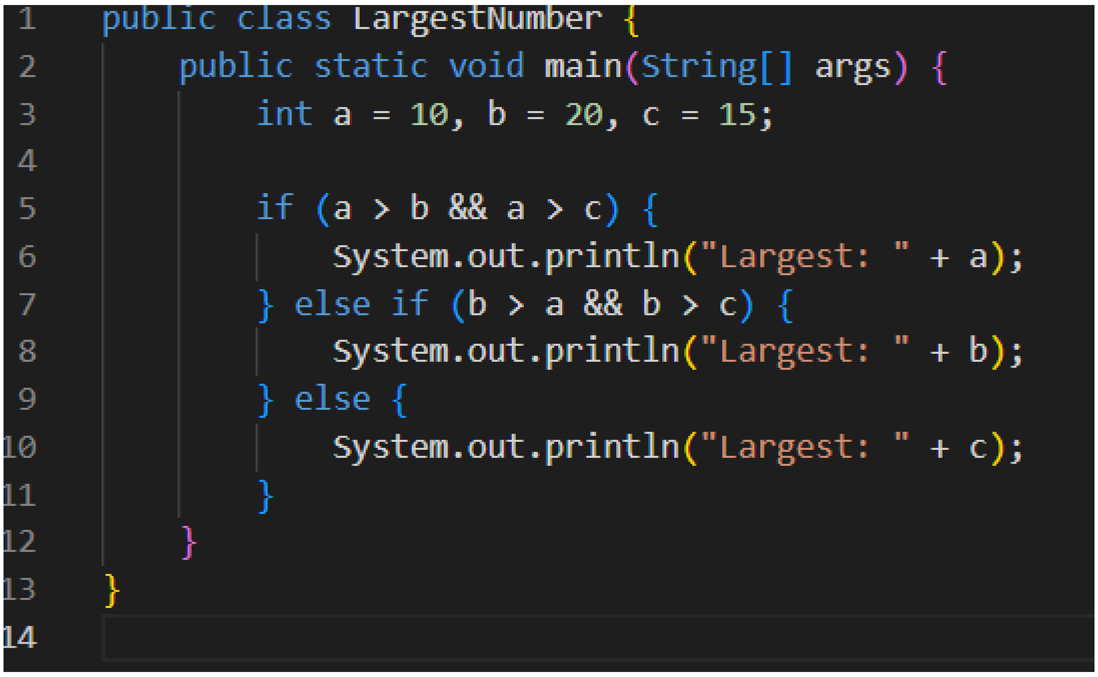
2.e) Deployment Diagram:



3.Basic Java Programs

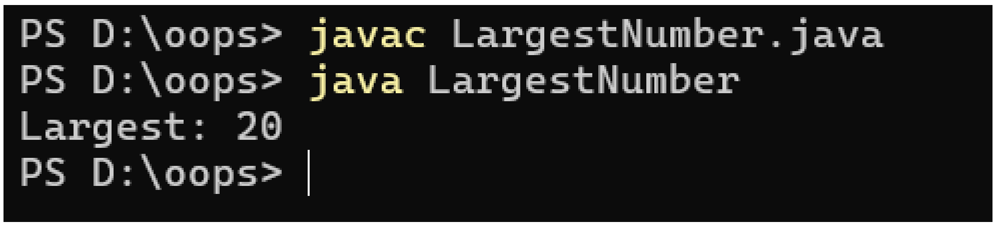
3. a) Largest Number:

Code:



```java
public class LargestNumber {
    public static void main(String[] args) {
        int a = 10, b = 20, c = 15;

        if (a > b && a > c) {
            System.out.println("Largest: " + a);
        } else if (b > a && b > c) {
            System.out.println("Largest: " + b);
        } else {
            System.out.println("Largest: " + c);
        }
    }
}
```

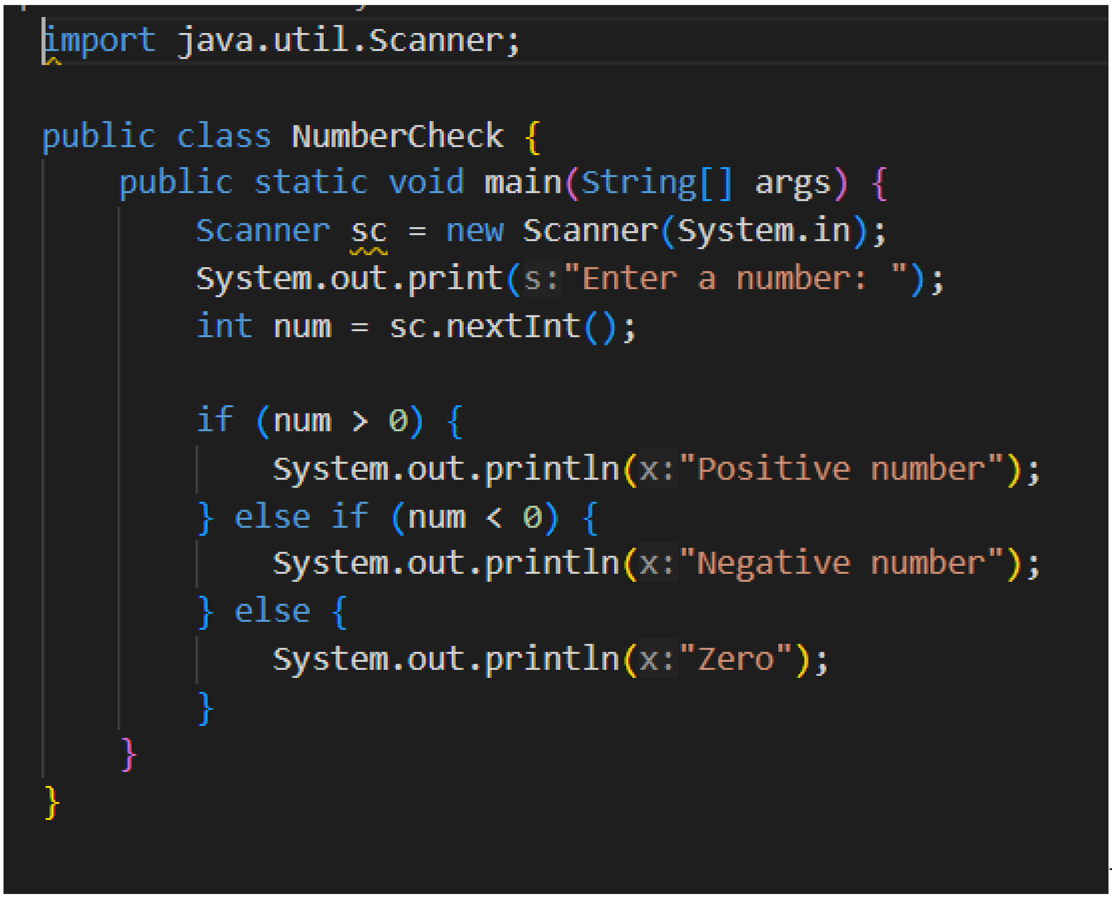
Output:



```
PS D:\oops> javac LargestNumber.java
PS D:\oops> java LargestNumber
Largest: 20
PS D:\oops>
```

3.b) Number Check:

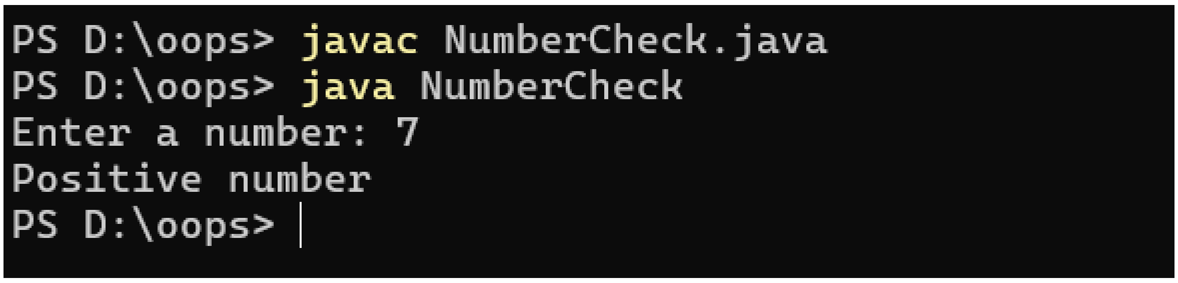
Code:



```java
import java.util.Scanner;

public class NumberCheck {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Enter a number: ");
        int num = sc.nextInt();

        if (num > 0) {
            System.out.println(x:"Positive number");
        } else if (num < 0) {
            System.out.println(x:"Negative number");
        } else {
            System.out.println(x:"Zero");
        }
    }
}
```

}

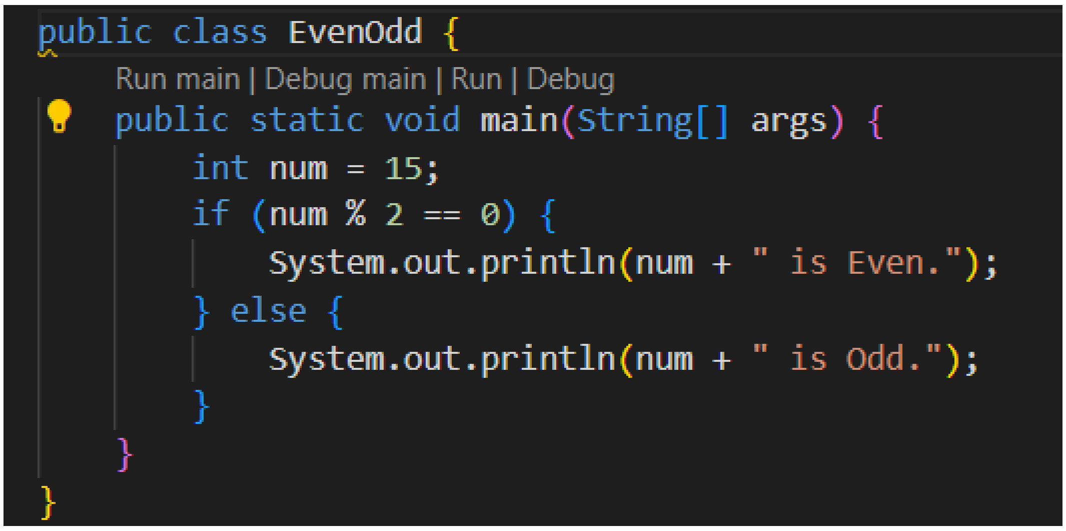
Output:



```
PS D:\oops> javac NumberCheck.java
PS D:\oops> java NumberCheck
Enter a number: 7
Positive number
PS D:\oops>
```

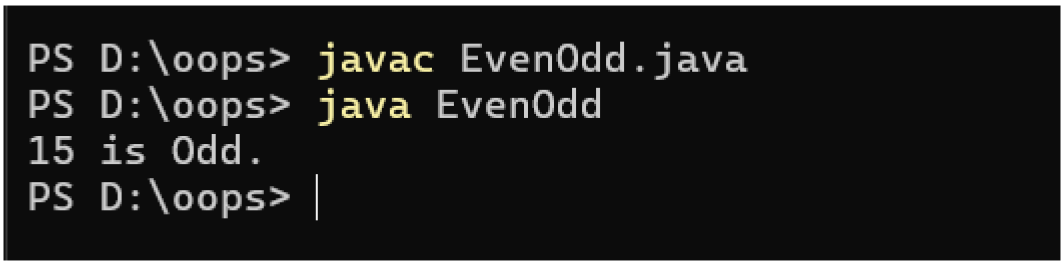
3.c) Even or Odd:

Code:



```
public class EvenOdd {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        int num = 15;
        if (num % 2 == 0) {
            System.out.println(num + " is Even.");
        } else {
            System.out.println(num + " is Odd.");
        }
    }
}
```

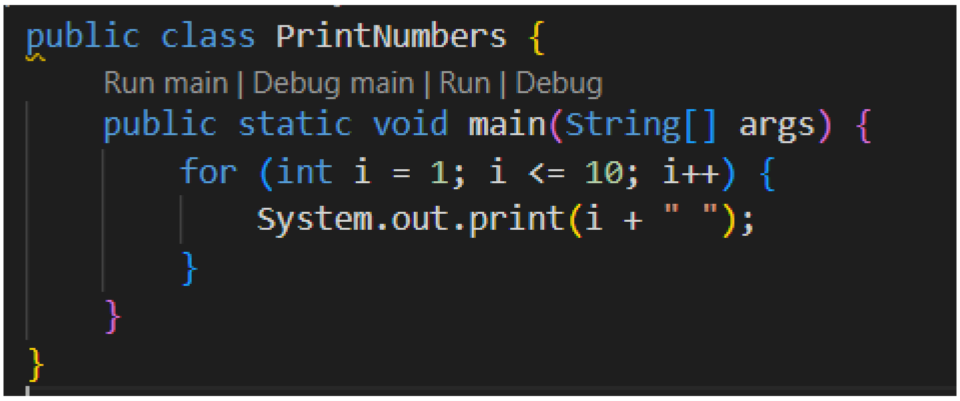
Output:



```
PS D:\oops> javac EvenOdd.java
PS D:\oops> java EvenOdd
15 is Odd.
PS D:\oops>
```

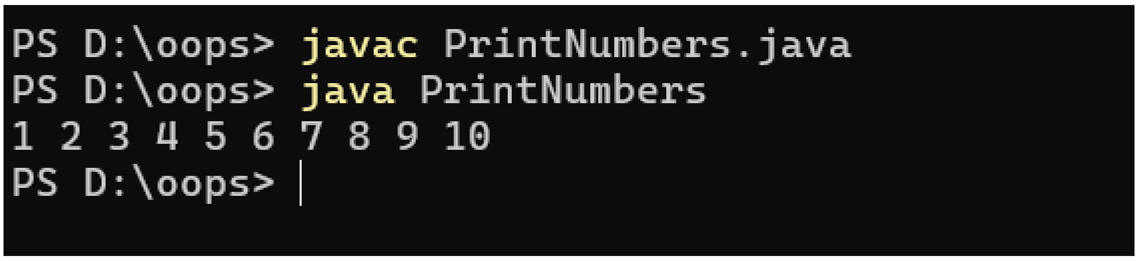
3.d) Print Numbers:

Code:



```
public class PrintNumbers {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        for (int i = 1; i <= 10; i++) {
            System.out.print(i + " ");
        }
    }
}
```

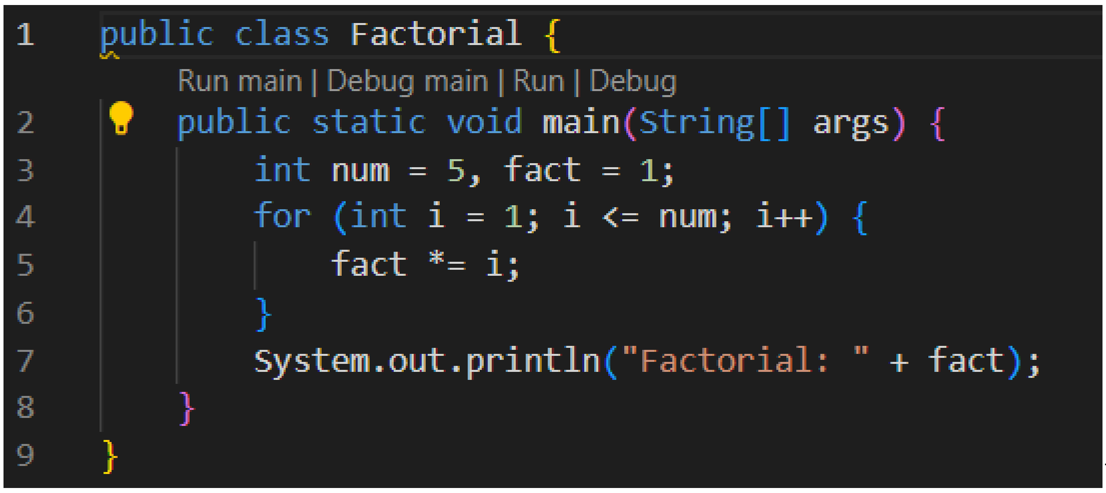
Output:



```
PS D:\oops> javac PrintNumbers.java
PS D:\oops> java PrintNumbers
1 2 3 4 5 6 7 8 9 10
PS D:\oops>
```

3.e) Factorial:

Code:

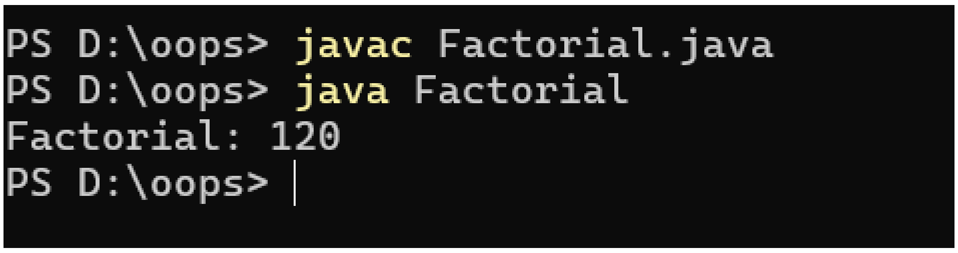


```java
public class Factorial {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        int num = 5, fact = 1;
        for (int i = 1; i <= num; i++) {
            fact *= i;
        }
        System.out.println("Factorial: " + fact);
    }
}
```

}

}

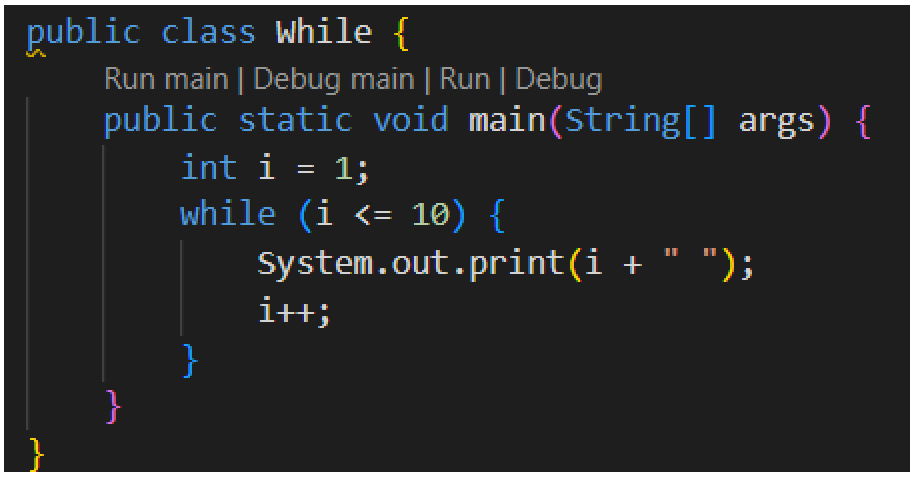
Output:



```
PS D:\oops> javac Factorial.java
PS D:\oops> java Factorial
Factorial: 120
PS D:\oops>
```

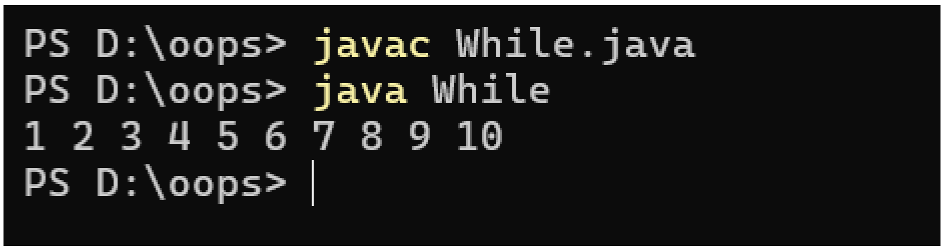
3.f) While:

Code:



```
public class While {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        int i = 1;
        while (i <= 10) {
            System.out.print(i + " ");
            i++;
        }
    }
}
```

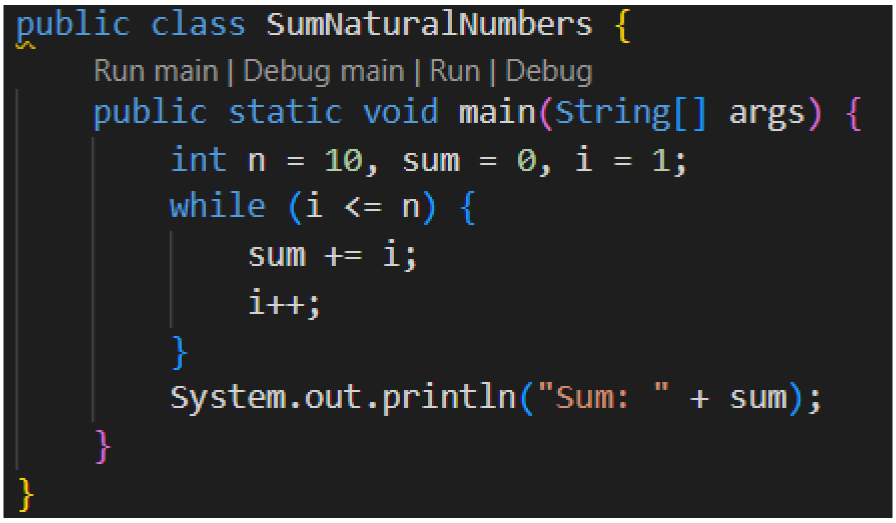
Output:



```
PS D:\oops> javac While.java
PS D:\oops> java While
1 2 3 4 5 6 7 8 9 10
PS D:\oops> 
```

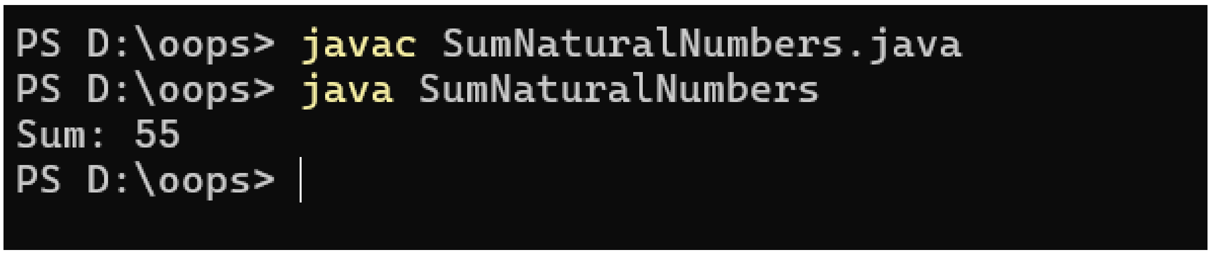
3.g) Sum Natural Numbers:

Code:



```
public class SumNaturalNumbers {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        int n = 10, sum = 0, i = 1;
        while (i <= n) {
            sum += i;
            i++;
        }
        System.out.println("Sum: " + sum);
    }
}
```

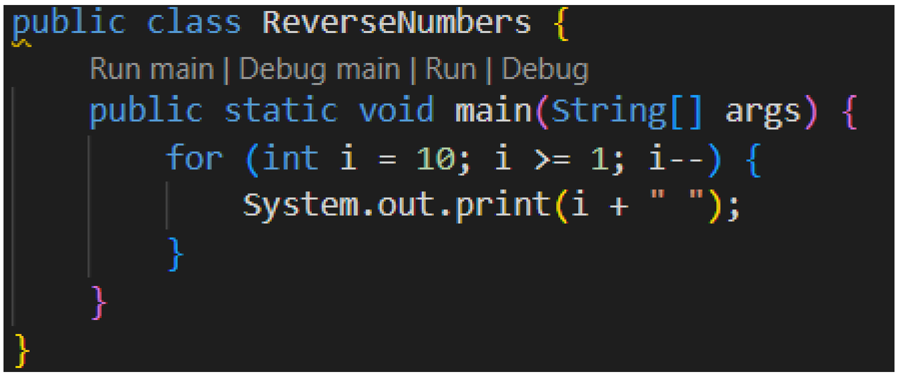
Output:



```
PS D:\oops> javac SumNaturalNumbers.java
PS D:\oops> java SumNaturalNumbers
Sum: 55
PS D:\oops> 
```

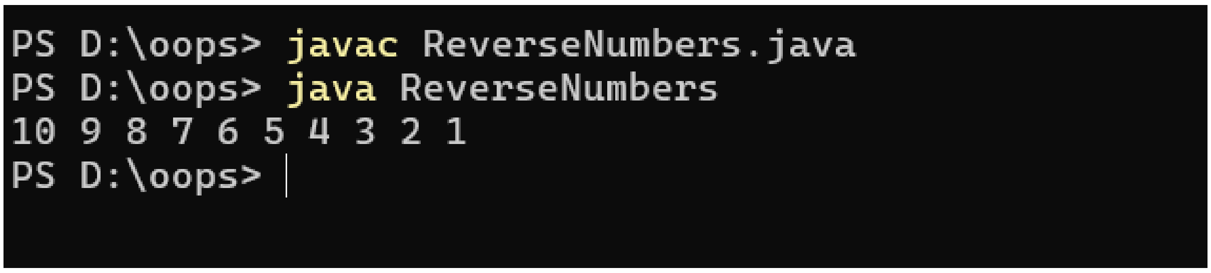
3.h) Reverse Numbers:

Code:



```
public class ReverseNumbers {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        for (int i = 10; i >= 1; i--) {
            System.out.print(i + " ");
        }
    }
}
```

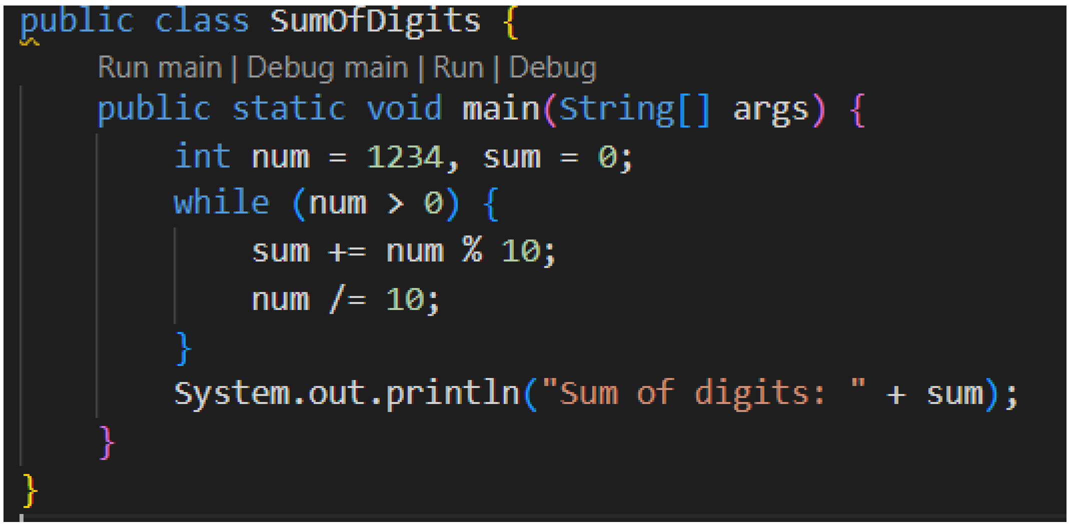
Output:



```
PS D:\oops> javac ReverseNumbers.java
PS D:\oops> java ReverseNumbers
10 9 8 7 6 5 4 3 2 1
PS D:\oops> 
```

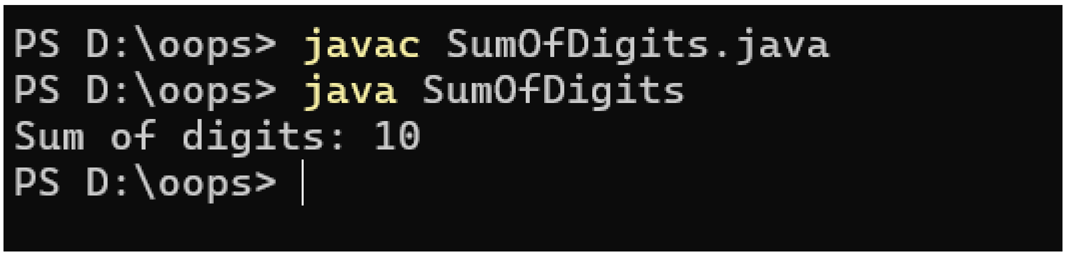
3.i) Sum Of Digits:

Code:



```
public class SumOfDigits {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        int num = 1234, sum = 0;
        while (num > 0) {
            sum += num % 10;
            num /= 10;
        }
        System.out.println("Sum of digits: " + sum);
    }
}
```

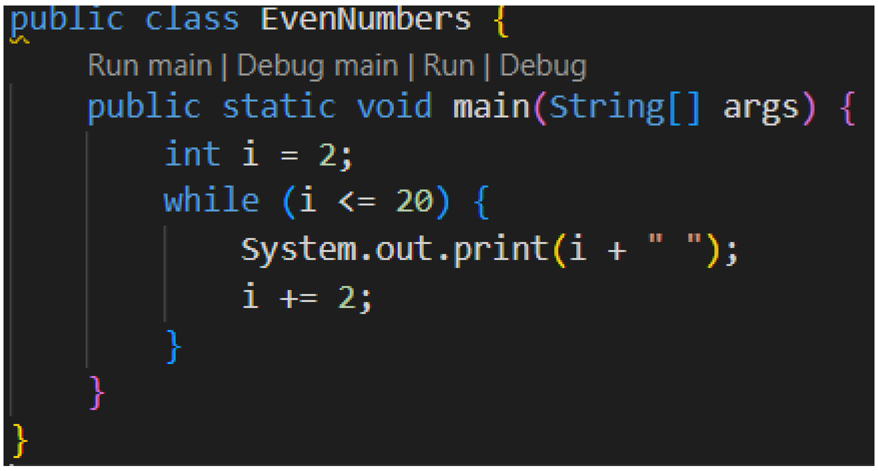
Output:



```
PS D:\oops> javac SumOfDigits.java
PS D:\oops> java SumOfDigits
Sum of digits: 10
PS D:\oops> 
```

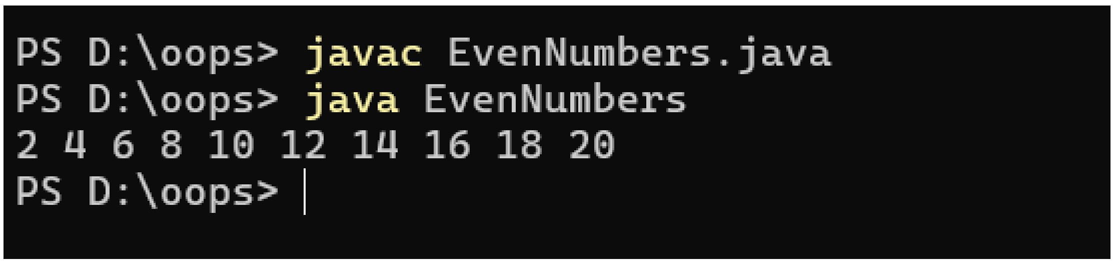
3.j) Even Numbers:

Code:



```
public class EvenNumbers {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        int i = 2;
        while (i <= 20) {
            System.out.print(i + " ");
            i += 2;
        }
    }
}
```

Output:



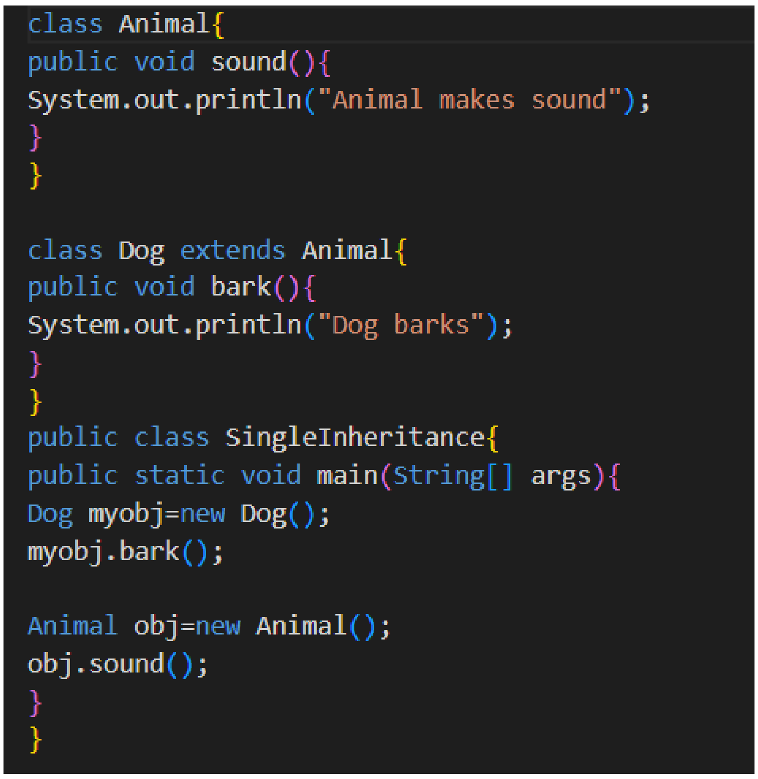
```
PS D:\oops> javac EvenNumbers.java
PS D:\oops> java EvenNumbers
2 4 6 8 10 12 14 16 18 20
PS D:\oops> 
```

INHERITANCE

4)SINGLE INHERITANCE PROGRAMS

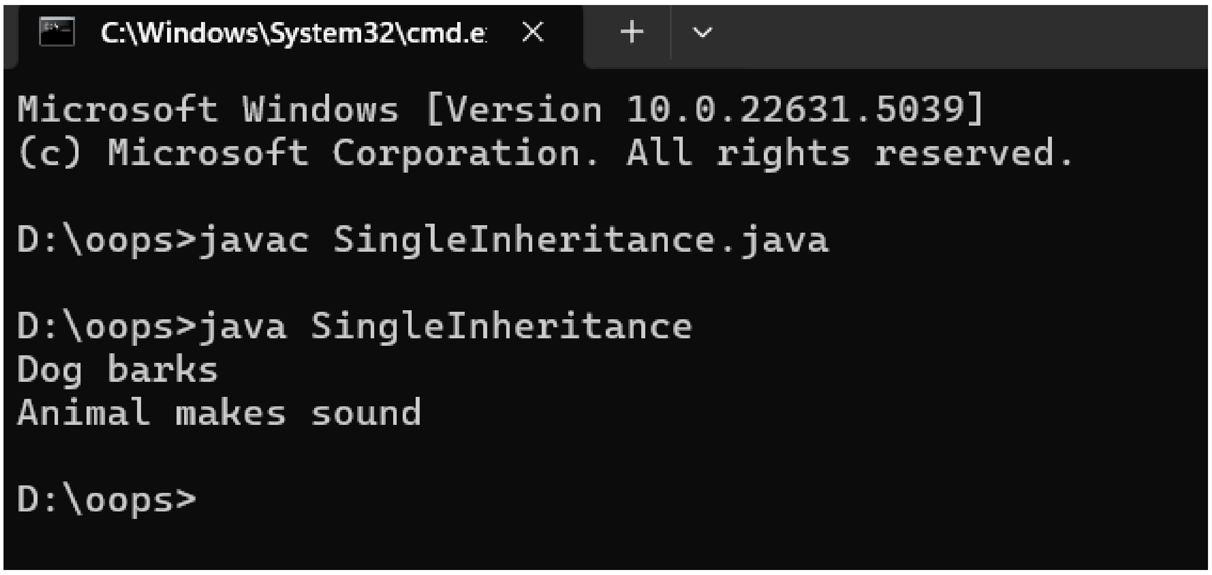
4a) Animal Sounds

Code:



```java
class Animal{
public void sound(){
System.out.println("Animal makes sound");
}
}

class Dog extends Animal{
public void bark(){
System.out.println("Dog barks");
}
}
public class SingleInheritance{
public static void main(String[] args){
Dog myobj=new Dog();
myobj.bark();

Animal obj=new Animal();
obj.sound();
}
}
```

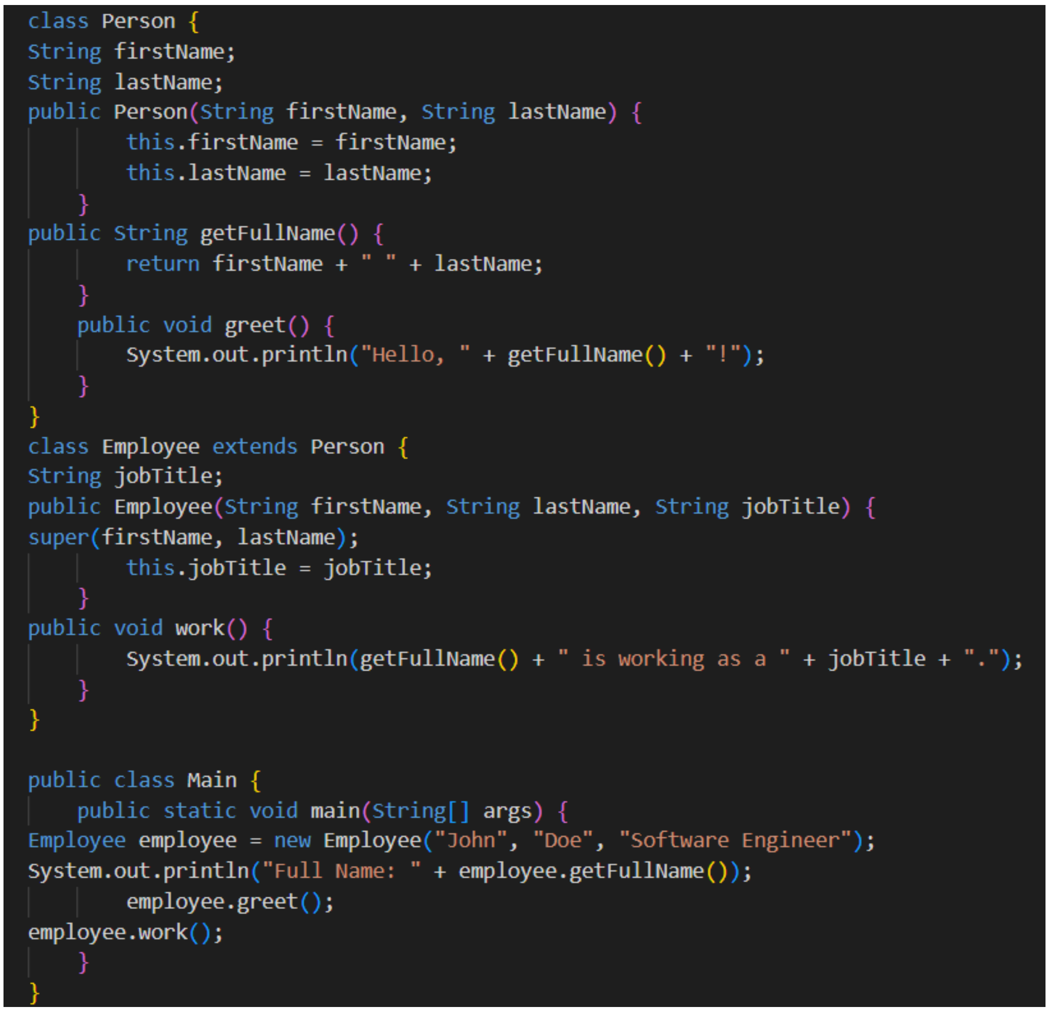
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac SingleInheritance.java

D:\oops>java SingleInheritance
Dog barks
Animal makes sound

D:\oops>
```

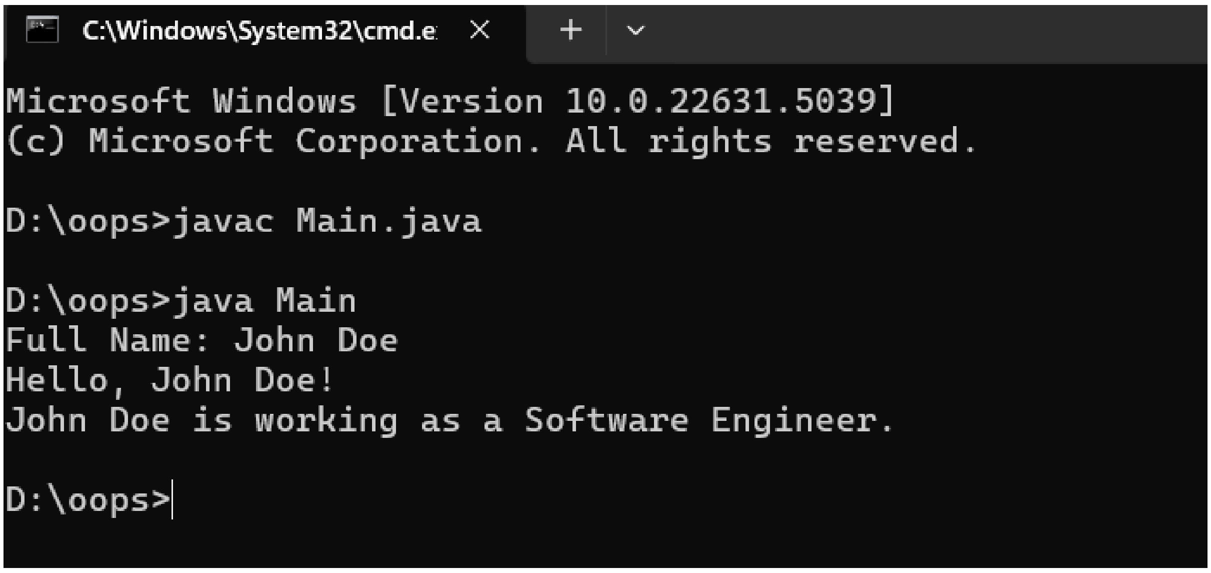
4b) Employee Details

Code:



```java
class Person {
String firstName;
String lastName;
public Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }
public String getFullName() {
        return firstName + " " + lastName;
    }
    public void greet() {
        System.out.println("Hello, " + getFullName() + "!");
    }
}
class Employee extends Person {
String jobTitle;
public Employee(String firstName, String lastName, String jobTitle) {
super(firstName, lastName);
        this.jobTitle = jobTitle;
    }
public void work() {
        System.out.println(getFullName() + " is working as a " + jobTitle + ".");
    }
}

public class Main {
    public static void main(String[] args) {
Employee employee = new Employee("John", "Doe", "Software Engineer");
System.out.println("Full Name: " + employee.getFullName());
        employee.greet();
employee.work();
    }
}
```

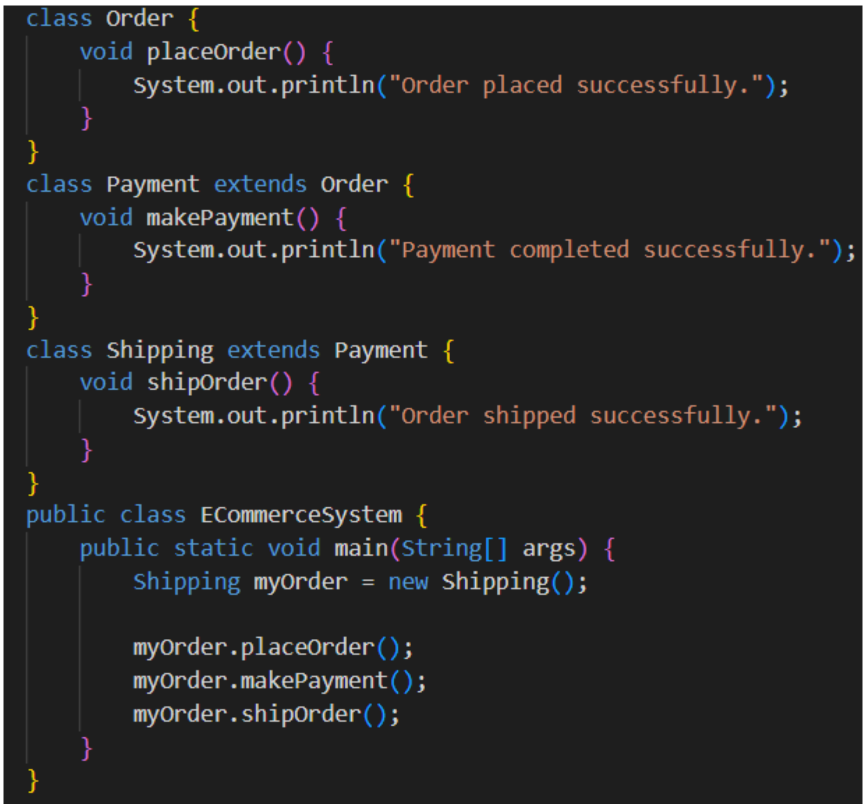
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac Main.java

D:\oops>java Main
Full Name: John Doe
Hello, John Doe!
John Doe is working as a Software Engineer.

D:\oops>
```

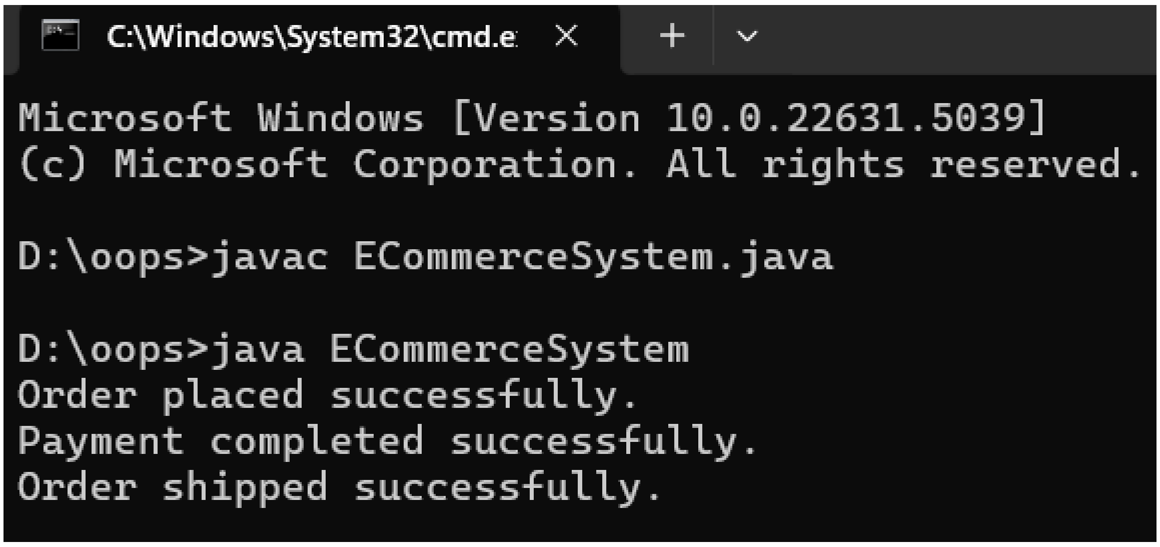
5) MULTILEVEL INHERITANCE PROGRAMS

5a) Ecommerce System



```java
class Order {
    void placeOrder() {
        System.out.println("Order placed successfully.");
    }
}
class Payment extends Order {
    void makePayment() {
        System.out.println("Payment completed successfully.");
    }
}
class Shipping extends Payment {
    void shipOrder() {
        System.out.println("Order shipped successfully.");
    }
}
public class ECommerceSystem {
    public static void main(String[] args) {
        Shipping myOrder = new Shipping();

        myOrder.placeOrder();
        myOrder.makePayment();
        myOrder.shipOrder();
    }
}
```

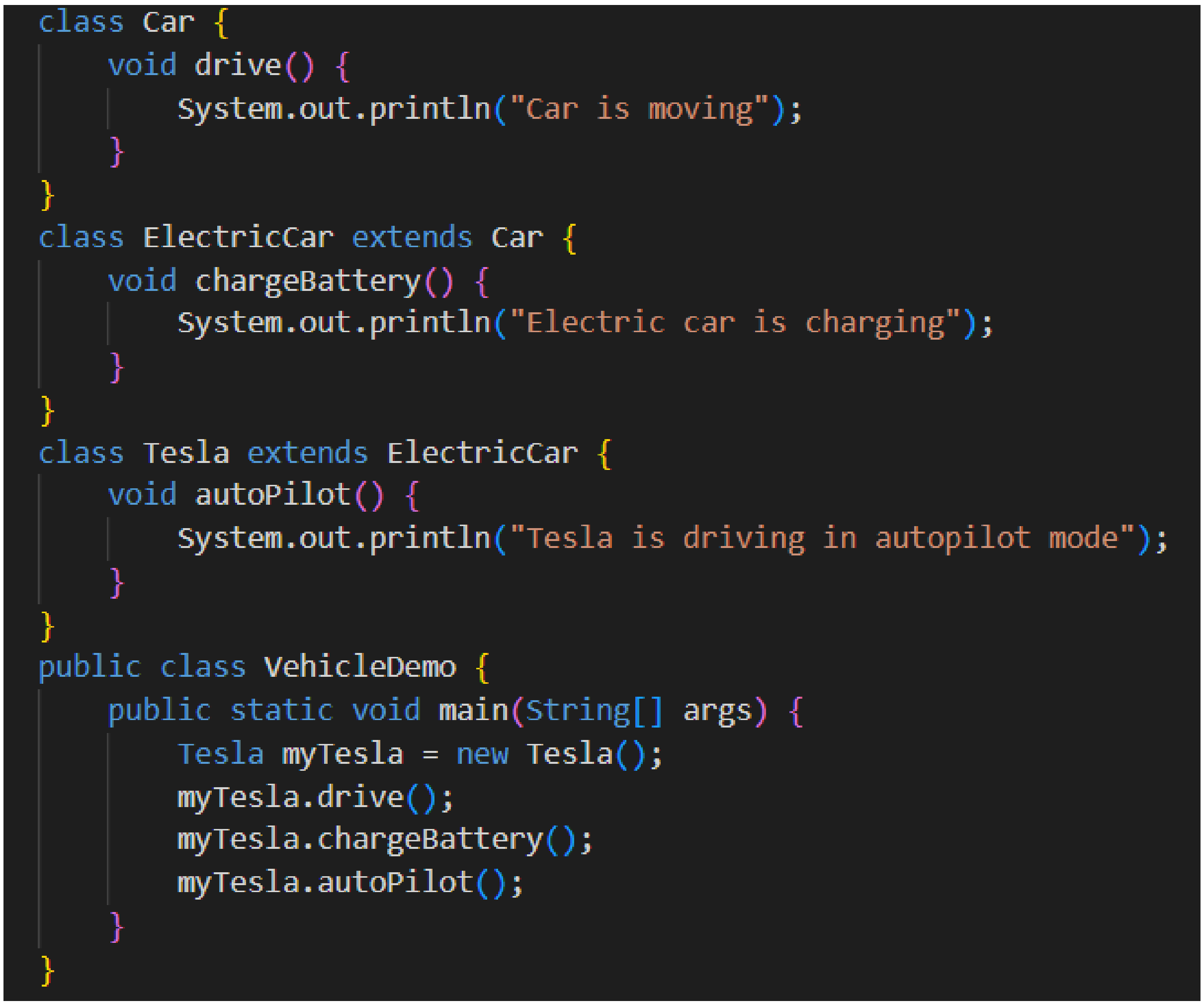
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac ECommerceSystem.java

D:\oops>java ECommerceSystem
Order placed successfully.
Payment completed successfully.
Order shipped successfully.
```

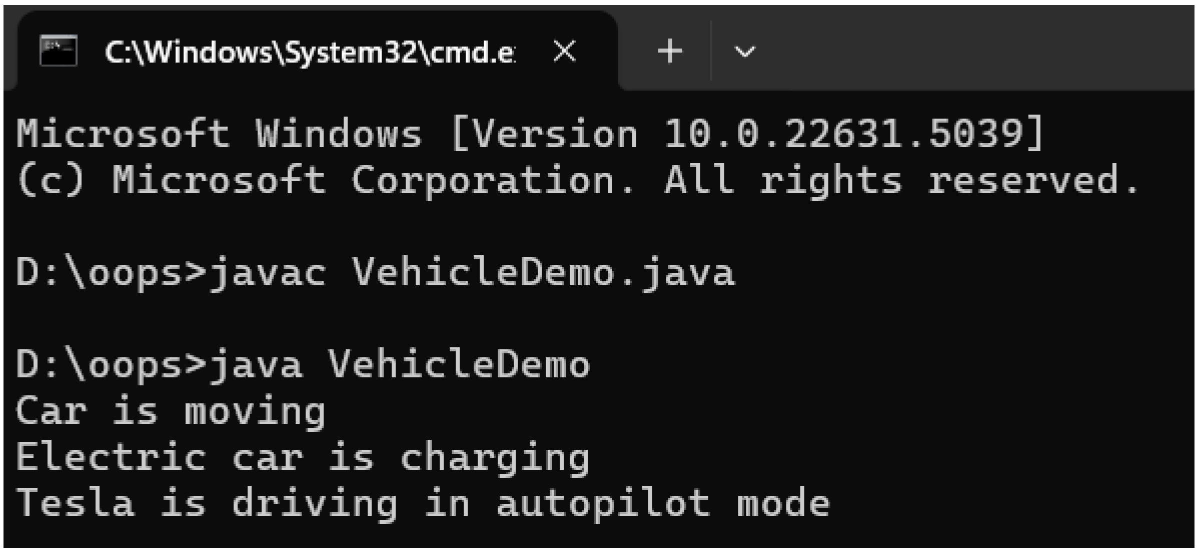
5b) Vehicledemo

Code:



```java
class Car {
    void drive() {
        System.out.println("Car is moving");
    }
}
class ElectricCar extends Car {
    void chargeBattery() {
        System.out.println("Electric car is charging");
    }
}
class Tesla extends ElectricCar {
    void autoPilot() {
        System.out.println("Tesla is driving in autopilot mode");
    }
}
public class VehicleDemo {
    public static void main(String[] args) {
        Tesla myTesla = new Tesla();
        myTesla.drive();
        myTesla.chargeBattery();
        myTesla.autoPilot();
    }
}
```

Output:

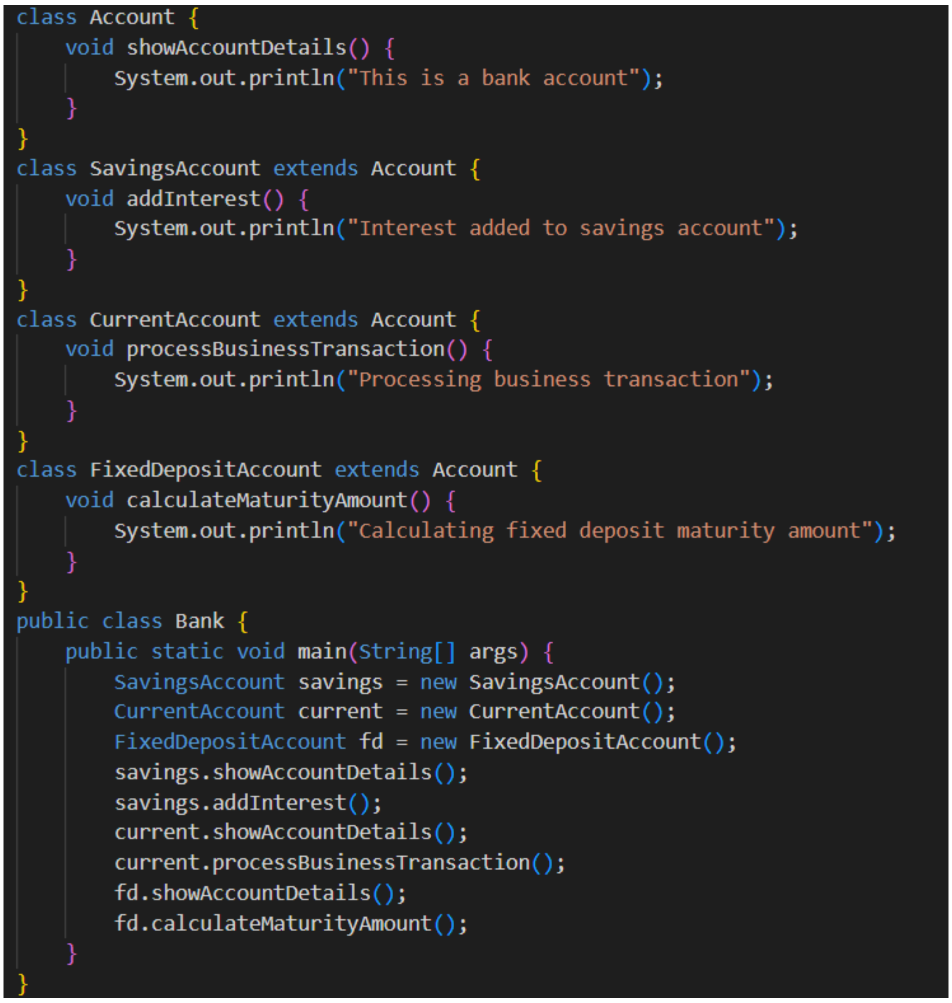


```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac VehicleDemo.java

D:\oops>java VehicleDemo
Car is moving
Electric car is charging
Tesla is driving in autopilot mode
```

6) HIERARCHICAL INHERITANCE PROGRAMS

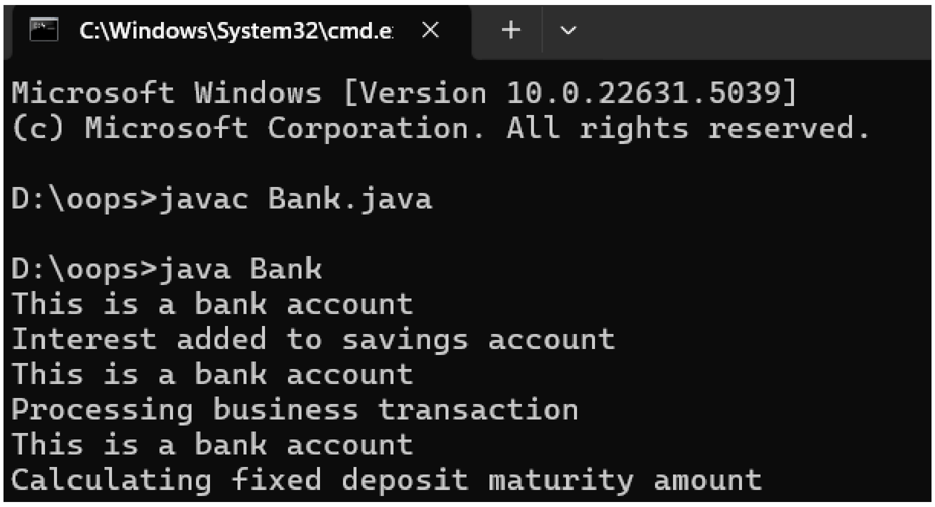
6a)Bank

Code:



```java
class Account {
    void showAccountDetails() {
        System.out.println("This is a bank account");
    }
}
class SavingsAccount extends Account {
    void addInterest() {
        System.out.println("Interest added to savings account");
    }
}
class CurrentAccount extends Account {
    void processBusinessTransaction() {
        System.out.println("Processing business transaction");
    }
}
class FixedDepositAccount extends Account {
    void calculateMaturityAmount() {
        System.out.println("Calculating fixed deposit maturity amount");
    }
}
public class Bank {
    public static void main(String[] args) {
        SavingsAccount savings = new SavingsAccount();
        CurrentAccount current = new CurrentAccount();
        FixedDepositAccount fd = new FixedDepositAccount();
        savings.showAccountDetails();
        savings.addInterest();
        current.showAccountDetails();
        current.processBusinessTransaction();
        fd.showAccountDetails();
        fd.calculateMaturityAmount();
    }
}
```

Output:

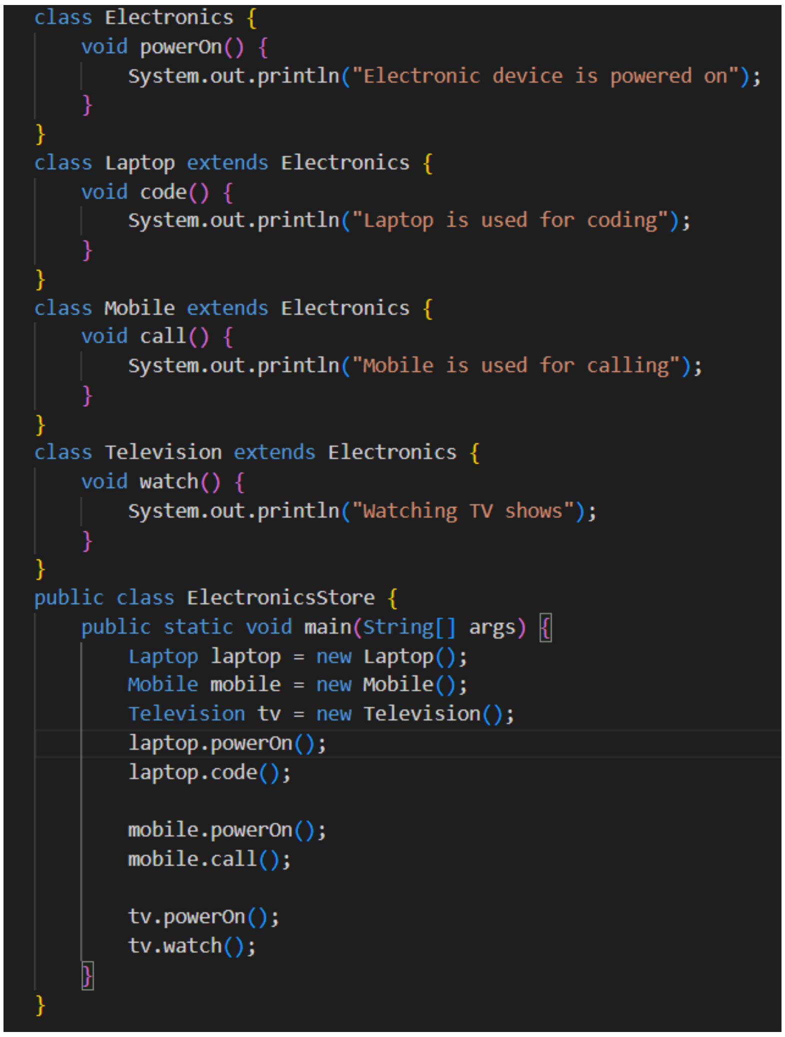


C:\Windows\System32\cmd.e

```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac Bank.java

D:\oops>java Bank
This is a bank account
Interest added to savings account
This is a bank account
Processing business transaction
This is a bank account
Calculating fixed deposit maturity amount
```

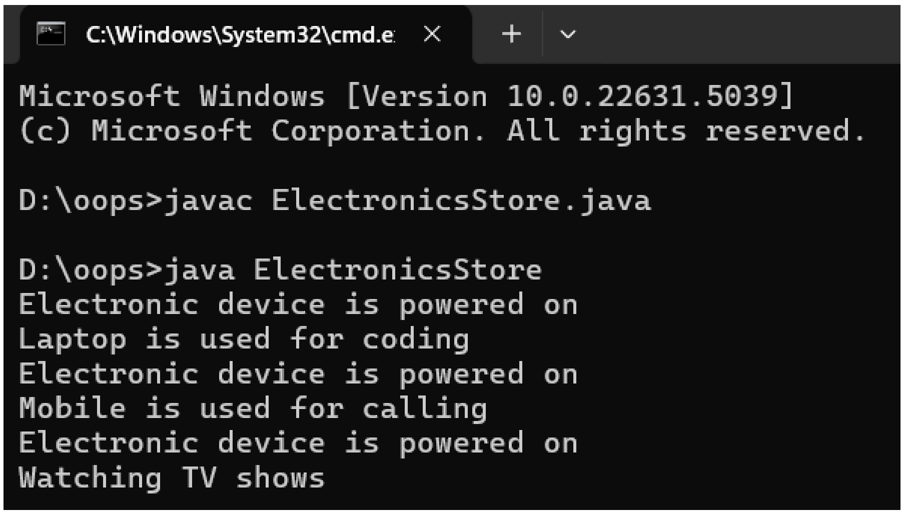
6b) ElectronicsStore

Code:



```java
class Electronics {
    void powerOn() {
        System.out.println("Electronic device is powered on");
    }
}
class Laptop extends Electronics {
    void code() {
        System.out.println("Laptop is used for coding");
    }
}
class Mobile extends Electronics {
    void call() {
        System.out.println("Mobile is used for calling");
    }
}
class Television extends Electronics {
    void watch() {
        System.out.println("Watching TV shows");
    }
}
public class ElectronicsStore {
    public static void main(String[] args) {
        Laptop laptop = new Laptop();
        Mobile mobile = new Mobile();
        Television tv = new Television();
        laptop.powerOn();
        laptop.code();

        mobile.powerOn();
        mobile.call();

        tv.powerOn();
        tv.watch();
    }
}
```

Output:

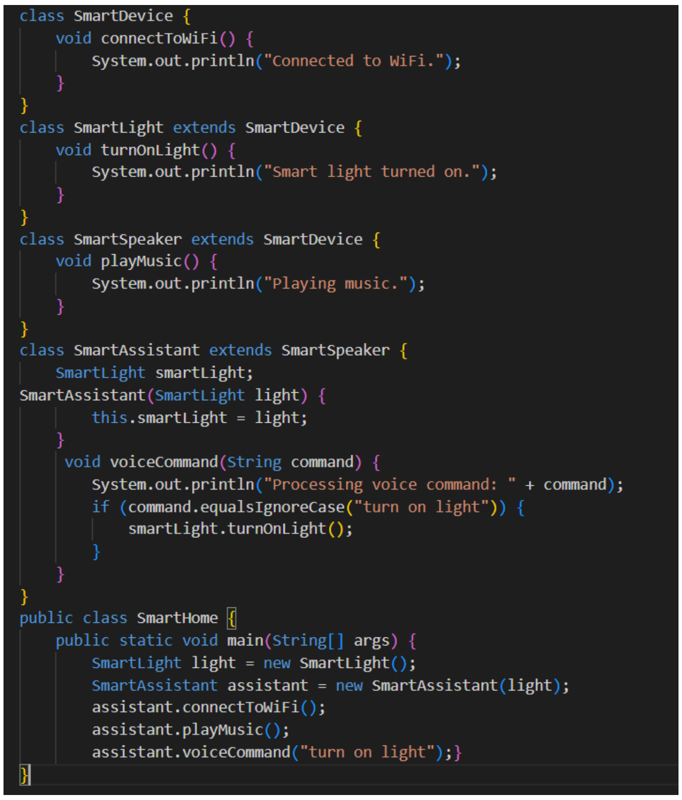


```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac ElectronicsStore.java

D:\oops>java ElectronicsStore
Electronic device is powered on
Laptop is used for coding
Electronic device is powered on
Mobile is used for calling
Electronic device is powered on
Watching TV shows
```

7) HYBRID INHERITANCE PROGRAMS

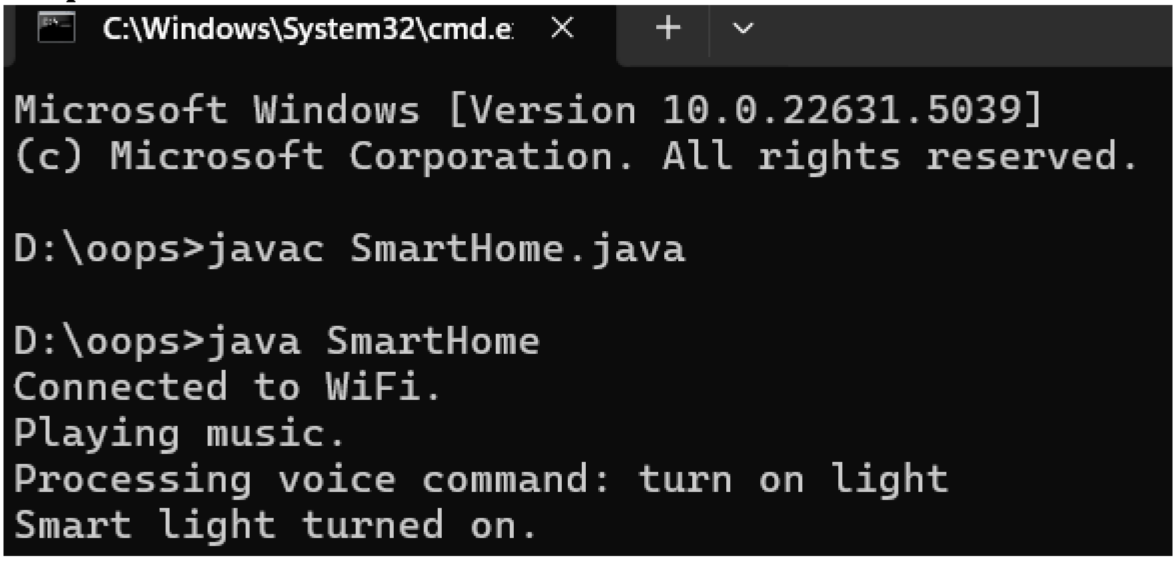
7a) SmartHome

Code:



```
class SmartDevice {
    void connectToWiFi() {
        System.out.println("Connected to WiFi.");
    }
}
class SmartLight extends SmartDevice {
    void turnOnLight() {
        System.out.println("Smart light turned on.");
    }
}
class SmartSpeaker extends SmartDevice {
    void playMusic() {
        System.out.println("Playing music.");
    }
}
class SmartAssistant extends SmartSpeaker {
    SmartLight smartLight;
SmartAssistant(SmartLight light) {
        this.smartLight = light;
    }
     void voiceCommand(String command) {
        System.out.println("Processing voice command: " + command);
        if (command.equalsIgnoreCase("turn on light")) {
            smartLight.turnOnLight();
        }
    }
}
public class SmartHome {
    public static void main(String[] args) {
        SmartLight light = new SmartLight();
        SmartAssistant assistant = new SmartAssistant(light);
        assistant.connectToWiFi();
        assistant.playMusic();
        assistant.voiceCommand("turn on light");}
}
```

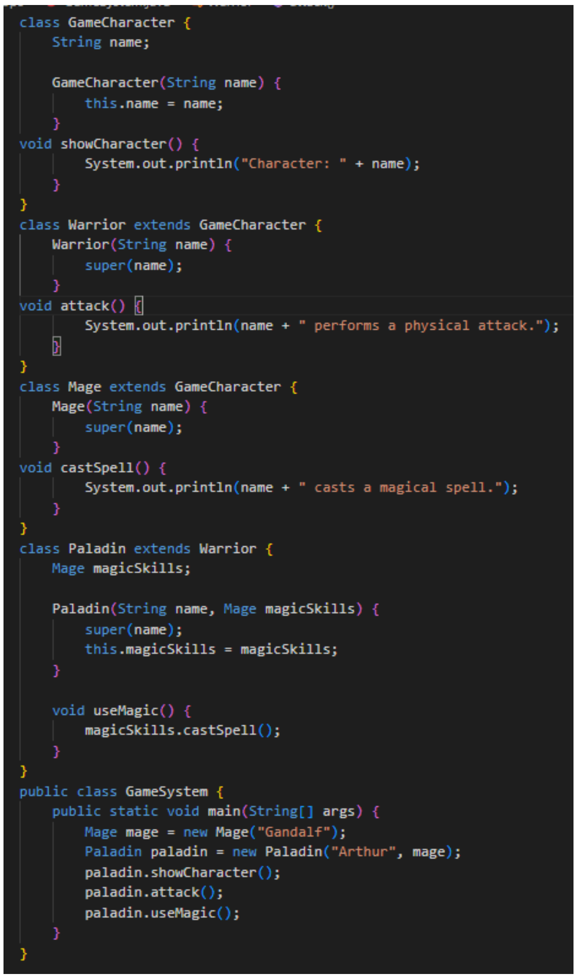
Output



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac SmartHome.java

D:\oops>java SmartHome
Connected to WiFi.
Playing music.
Processing voice command: turn on light
Smart light turned on.
```

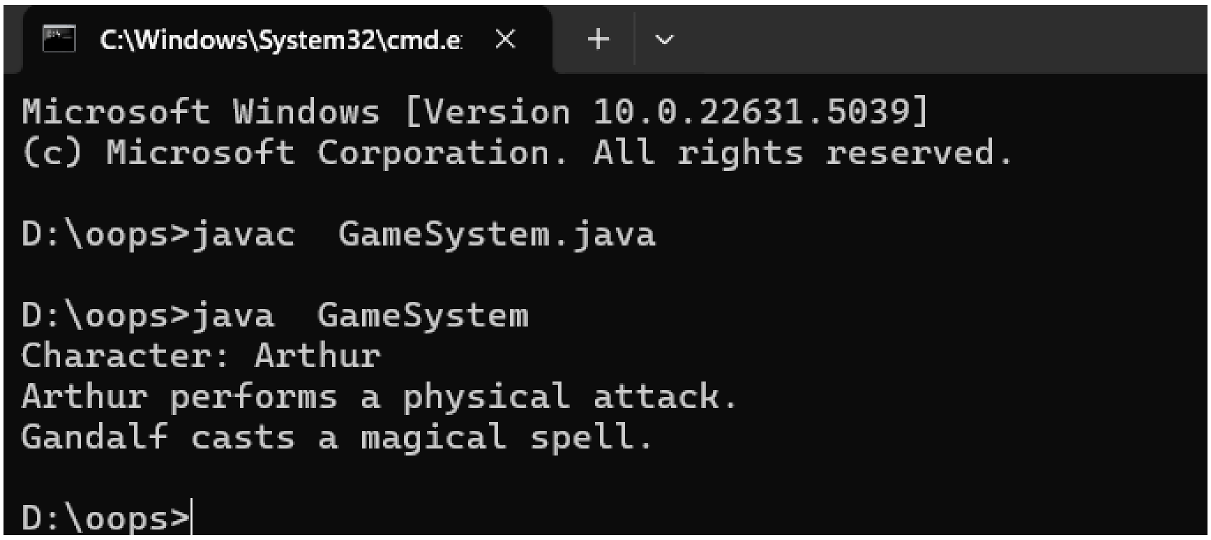
7b) GameSystem

Code:



```
class GameCharacter {
    String name;

    GameCharacter(String name) {
        this.name = name;
    }
void showCharacter() {
        System.out.println("Character: " + name);
    }
}
class Warrior extends GameCharacter {
    Warrior(String name) {
        super(name);
    }
void attack() {
        System.out.println(name + " performs a physical attack.");
    }
}
class Mage extends GameCharacter {
    Mage(String name) {
        super(name);
    }
void castSpell() {
        System.out.println(name + " casts a magical spell.");
    }
}
class Paladin extends Warrior {
    Mage magicSkills;

    Paladin(String name, Mage magicSkills) {
        super(name);
        this.magicSkills = magicSkills;
    }

    void useMagic() {
        magicSkills.castSpell();
    }
}
public class GameSystem {
    public static void main(String[] args) {
        Mage mage = new Mage("Gandalf");
        Paladin paladin = new Paladin("Arthur", mage);
        paladin.showCharacter();
        paladin.attack();
        paladin.useMagic();
    }
}
```

Output:



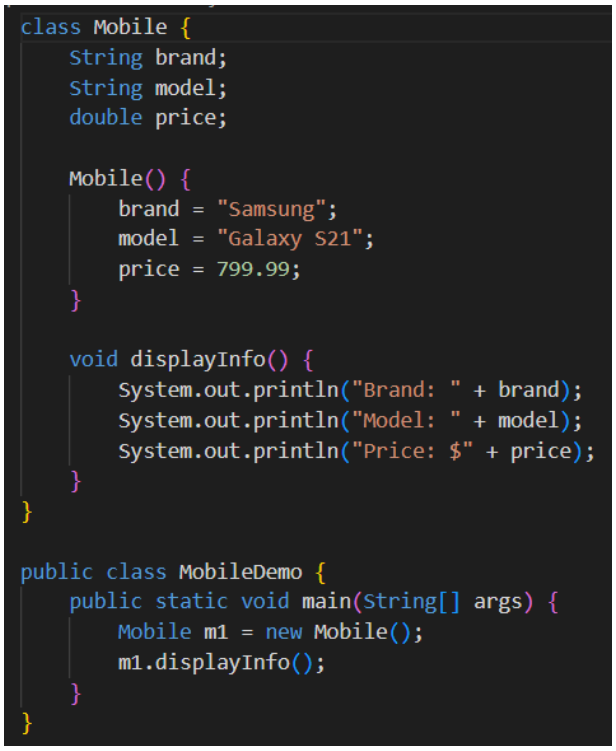
```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac  GameSystem.java

D:\oops>java  GameSystem
Character: Arthur
Arthur performs a physical attack.
Gandalf casts a magical spell.

D:\oops>
```

POLYMORPHISM

8) CONSTRUCTOR PROGRAMS

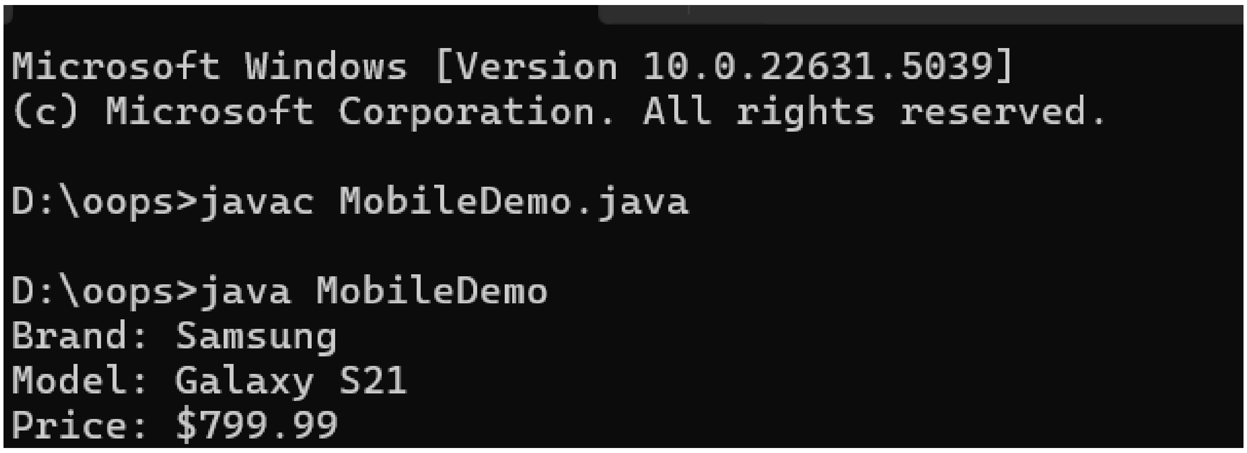
a) MobileDemo

Code:



```java
class Mobile {
    String brand;
    String model;
    double price;

    Mobile() {
        brand = "Samsung";
        model = "Galaxy S21";
        price = 799.99;
    }

    void displayInfo() {
        System.out.println("Brand: " + brand);
        System.out.println("Model: " + model);
        System.out.println("Price: $" + price);
    }
}

public class MobileDemo {
    public static void main(String[] args) {
        Mobile m1 = new Mobile();
        m1.displayInfo();
    }
}
```

Output:

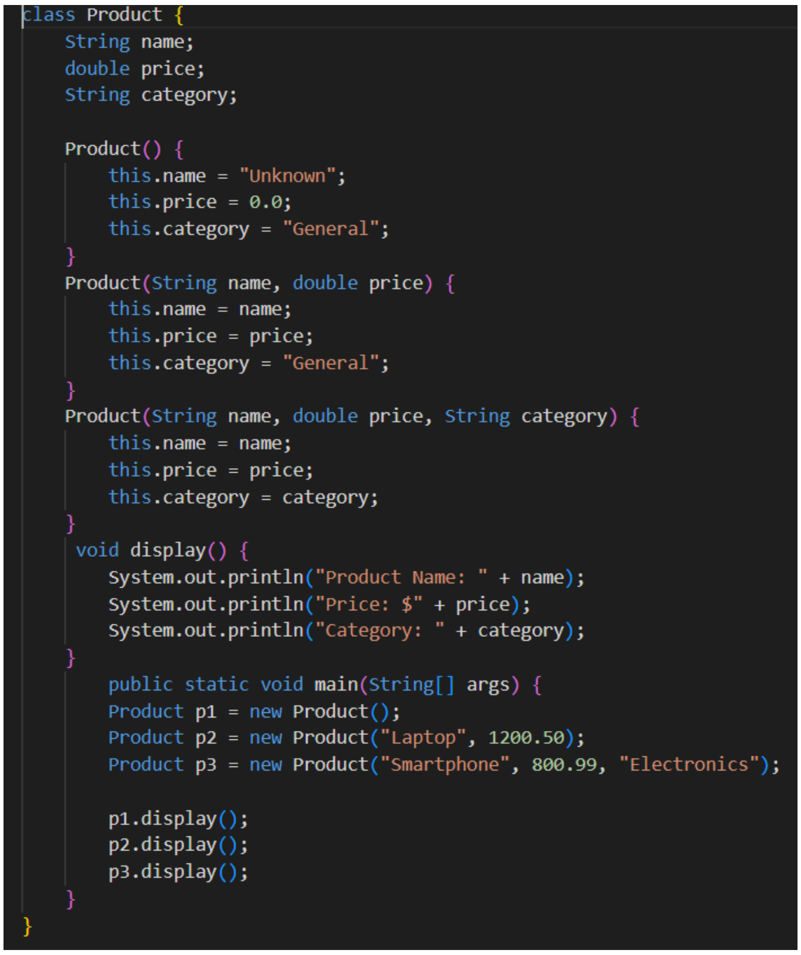


```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac MobileDemo.java

D:\oops>java MobileDemo
Brand: Samsung
Model: Galaxy S21
Price: $799.99
```

9)CONSTRUCTOR OVERLOADING PROGRAMS

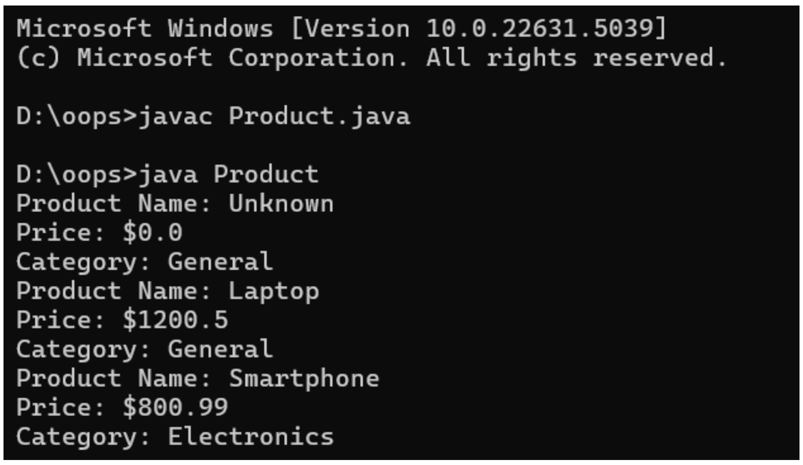
9.a)Product

Code:



```java
class Product {
    String name;
    double price;
    String category;

    Product() {
        this.name = "Unknown";
        this.price = 0.0;
        this.category = "General";
    }
    Product(String name, double price) {
        this.name = name;
        this.price = price;
        this.category = "General";
    }
    Product(String name, double price, String category) {
        this.name = name;
        this.price = price;
        this.category = category;
    }
     void display() {
        System.out.println("Product Name: " + name);
        System.out.println("Price: $" + price);
        System.out.println("Category: " + category);
    }
        public static void main(String[] args) {
        Product p1 = new Product();
        Product p2 = new Product("Laptop", 1200.50);
        Product p3 = new Product("Smartphone", 800.99, "Electronics");

        p1.display();
        p2.display();
        p3.display();
    }
}
```

Output:

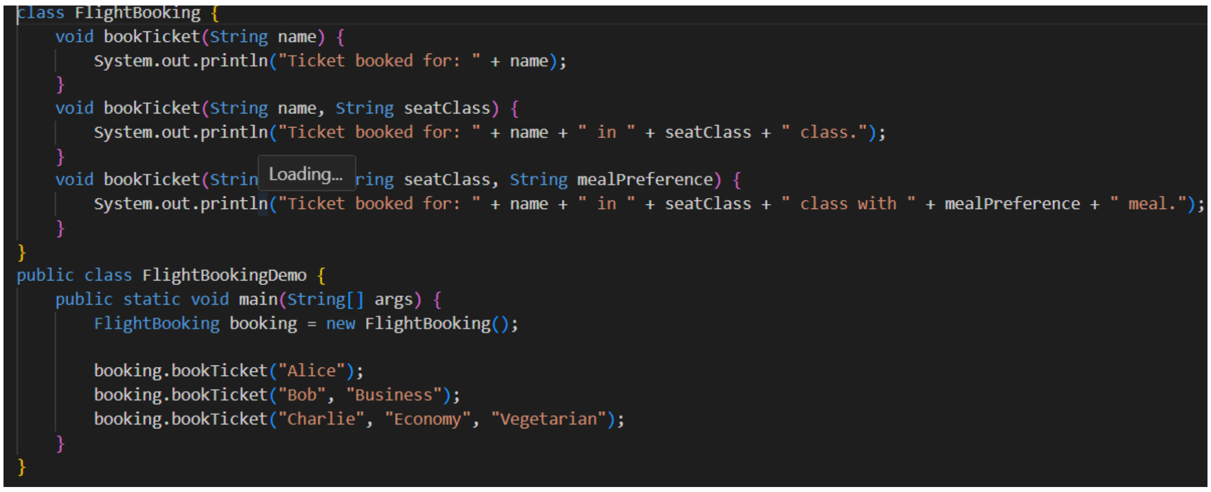


```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac Product.java

D:\oops>java Product
Product Name: Unknown
Price: $0.0
Category: General
Product Name: Laptop
Price: $1200.5
Category: General
Product Name: Smartphone
Price: $800.99
Category: Electronics
```

10)METHOD OVERLOADING PROGRAMS

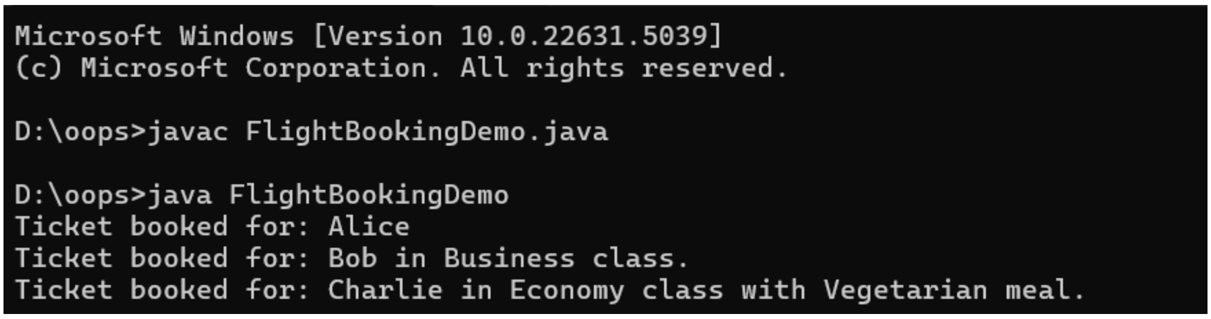
10.a) FlightBookingDemo

Code:



```
class FlightBooking {
    void bookTicket(String name) {
        System.out.println("Ticket booked for: " + name);
    }
    void bookTicket(String name, String seatClass) {
        System.out.println("Ticket booked for: " + name + " in " + seatClass + " class.");
    }
    void bookTicket(Strin Loading... ring seatClass, String mealPreference) {
        System.out.println("Ticket booked for: " + name + " in " + seatClass + " class with " + mealPreference + " meal.");
    }
}
public class FlightBookingDemo {
    public static void main(String[] args) {
        FlightBooking booking = new FlightBooking();

        booking.bookTicket("Alice");
        booking.bookTicket("Bob", "Business");
        booking.bookTicket("Charlie", "Economy", "Vegetarian");
    }
}
```

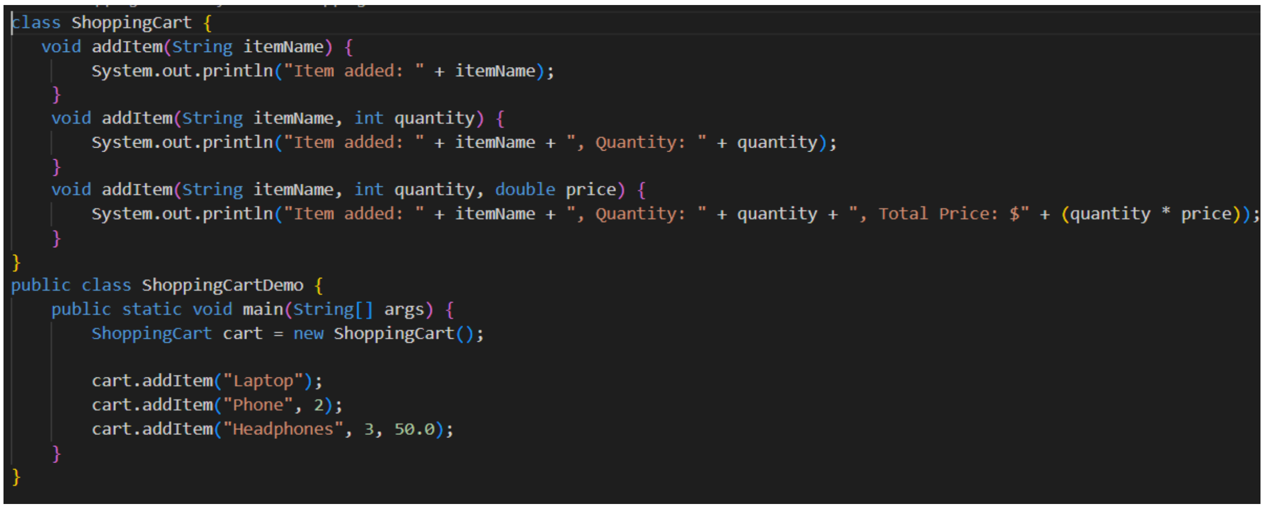
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac FlightBookingDemo.java

D:\oops>java FlightBookingDemo
Ticket booked for: Alice
Ticket booked for: Bob in Business class.
Ticket booked for: Charlie in Economy class with Vegetarian meal.
```

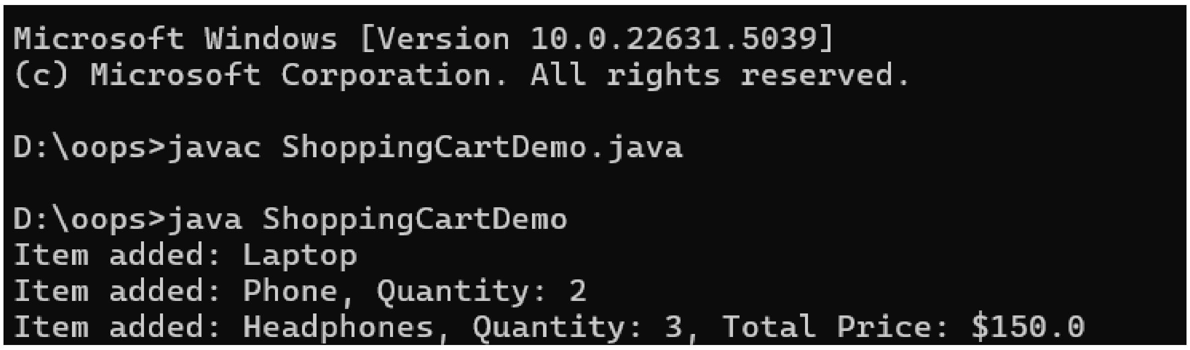
10.b)ShoppingCart:

Code:



```
class ShoppingCart {
    void addItem(String itemName) {
        System.out.println("Item added: " + itemName);
    }
    void addItem(String itemName, int quantity) {
        System.out.println("Item added: " + itemName + ", Quantity: " + quantity);
    }
    void addItem(String itemName, int quantity, double price) {
        System.out.println("Item added: " + itemName + ", Quantity: " + quantity + ", Total Price: $" + (quantity * price));
    }
}
public class ShoppingCartDemo {
    public static void main(String[] args) {
        ShoppingCart cart = new ShoppingCart();

        cart.addItem("Laptop");
        cart.addItem("Phone", 2);
        cart.addItem("Headphones", 3, 50.0);
    }
}
```

Output:

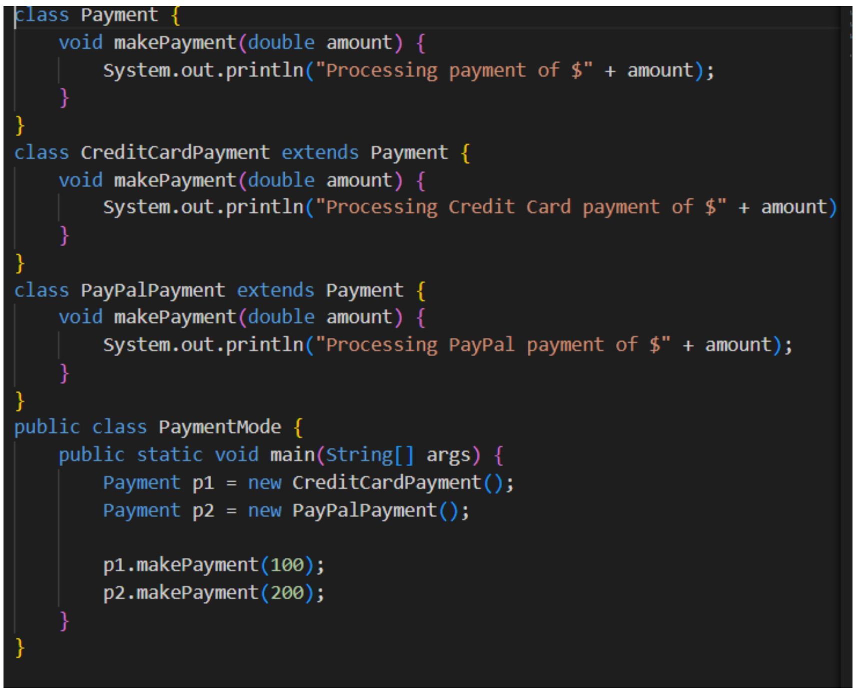


```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac ShoppingCartDemo.java

D:\oops>java ShoppingCartDemo
Item added: Laptop
Item added: Phone, Quantity: 2
Item added: Headphones, Quantity: 3, Total Price: $150.0
```

11) METHOD OVERRIDING PROGRAMS

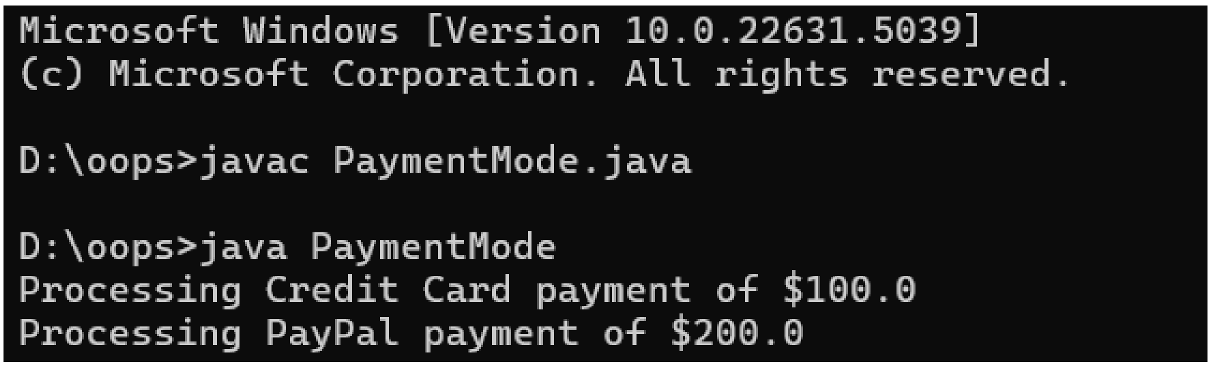
11.a) PaymentMode

Code:



```java
class Payment {
    void makePayment(double amount) {
        System.out.println("Processing payment of $" + amount);
    }
}
class CreditCardPayment extends Payment {
    void makePayment(double amount) {
        System.out.println("Processing Credit Card payment of $" + amount)
    }
}
class PayPalPayment extends Payment {
    void makePayment(double amount) {
        System.out.println("Processing PayPal payment of $" + amount);
    }
}
public class PaymentMode {
    public static void main(String[] args) {
        Payment p1 = new CreditCardPayment();
        Payment p2 = new PayPalPayment();

        p1.makePayment(100);
        p2.makePayment(200);
    }
}
```

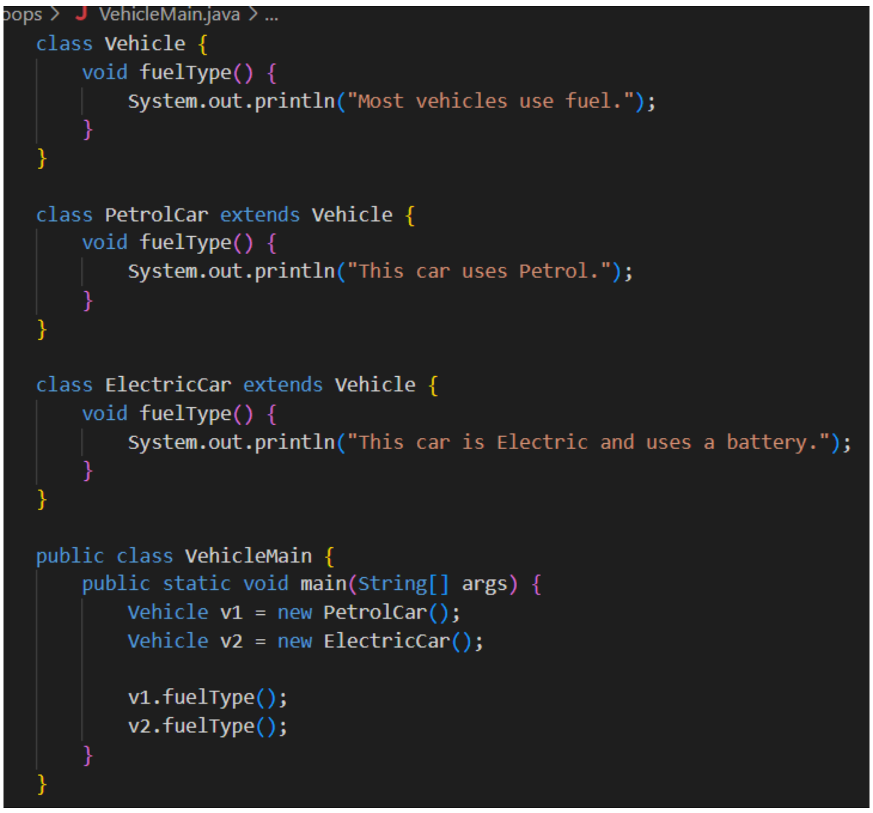
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac PaymentMode.java

D:\oops>java PaymentMode
Processing Credit Card payment of $100.0
Processing PayPal payment of $200.0
```

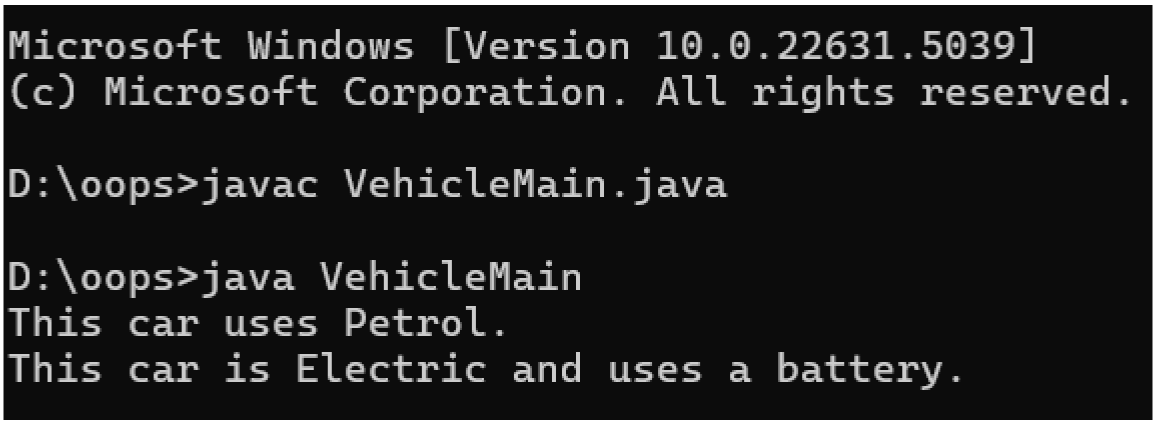
11.b) Vehicle

Code:



```
oops > VehicleMain.java > ...
class Vehicle {
    void fuelType() {
        System.out.println("Most vehicles use fuel.");
    }
}

class PetrolCar extends Vehicle {
    void fuelType() {
        System.out.println("This car uses Petrol.");
    }
}

class ElectricCar extends Vehicle {
    void fuelType() {
        System.out.println("This car is Electric and uses a battery.");
    }
}

public class VehicleMain {
    public static void main(String[] args) {
        Vehicle v1 = new PetrolCar();
        Vehicle v2 = new ElectricCar();

        v1.fuelType();
        v2.fuelType();
    }
}
```

Output:



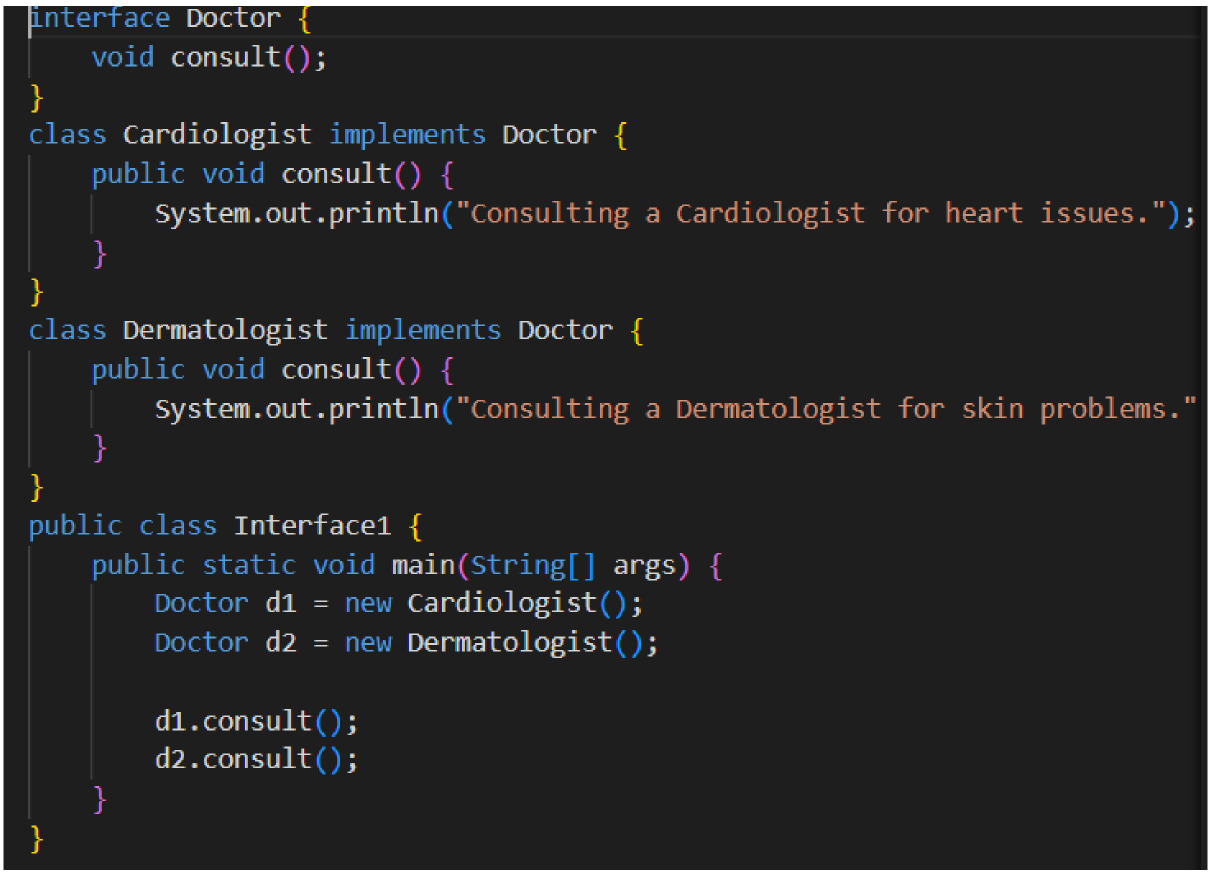
```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac VehicleMain.java

D:\oops>java VehicleMain
This car uses Petrol.
This car is Electric and uses a battery.
```

ABSTRACTION

12) INTERFACE PROGRAMS

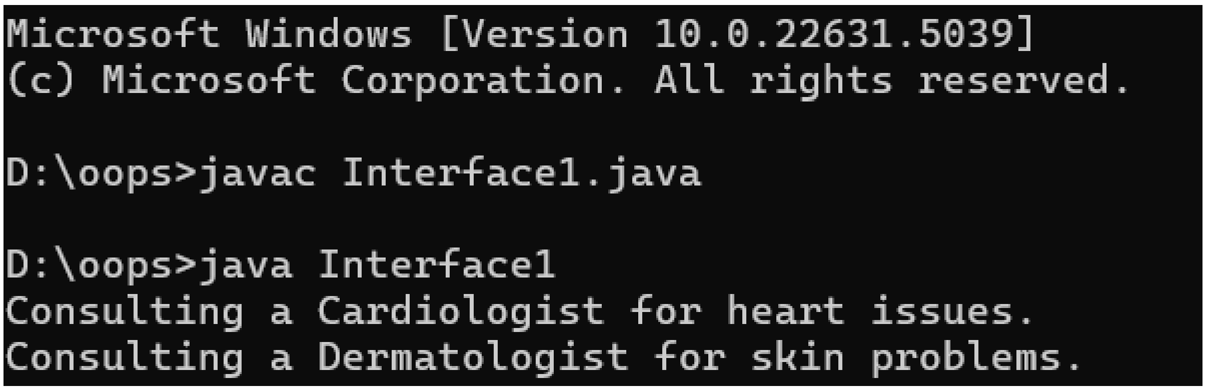
12a)Doctor

Code:



```
interface Doctor {
    void consult();
}
class Cardiologist implements Doctor {
    public void consult() {
        System.out.println("Consulting a Cardiologist for heart issues.");
    }
}
class Dermatologist implements Doctor {
    public void consult() {
        System.out.println("Consulting a Dermatologist for skin problems."
    }
}
public class Interface1 {
    public static void main(String[] args) {
        Doctor d1 = new Cardiologist();
        Doctor d2 = new Dermatologist();

        d1.consult();
        d2.consult();
    }
}
```

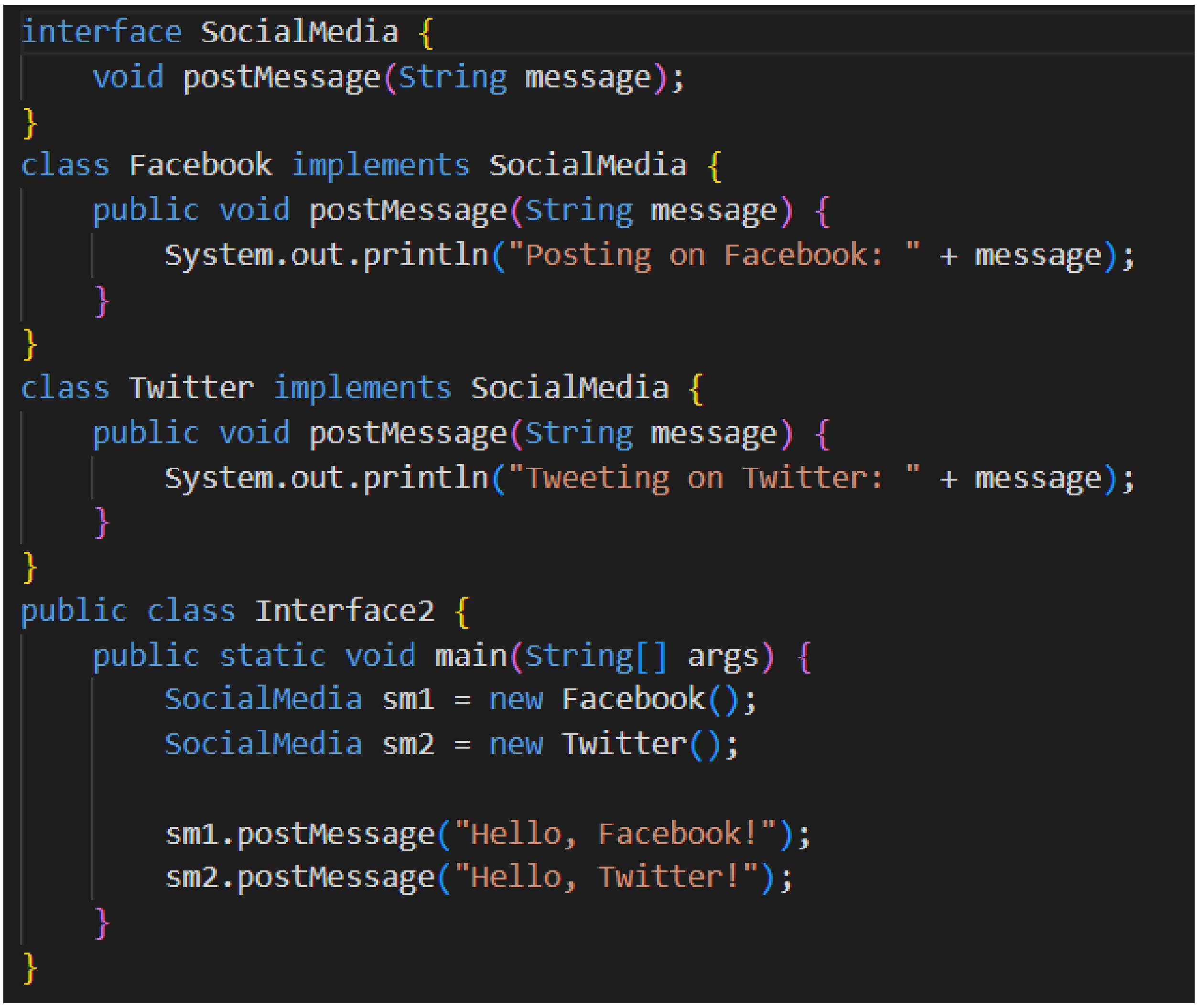
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac Interface1.java

D:\oops>java Interface1
Consulting a Cardiologist for heart issues.
Consulting a Dermatologist for skin problems.
```

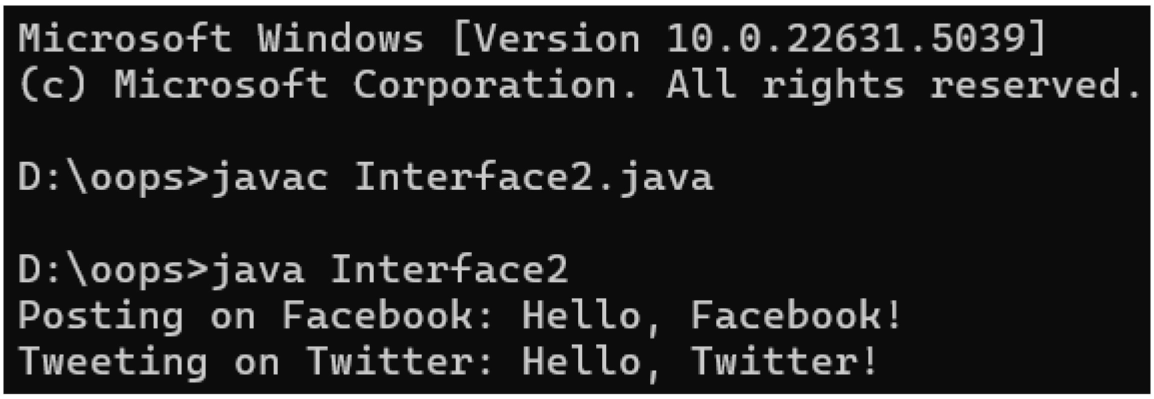
12b)SocialMedia

Code:



```
interface SocialMedia {
    void postMessage(String message);
}
class Facebook implements SocialMedia {
    public void postMessage(String message) {
        System.out.println("Posting on Facebook: " + message);
    }
}
class Twitter implements SocialMedia {
    public void postMessage(String message) {
        System.out.println("Tweeting on Twitter: " + message);
    }
}
public class Interface2 {
    public static void main(String[] args) {
        SocialMedia sm1 = new Facebook();
        SocialMedia sm2 = new Twitter();

        sm1.postMessage("Hello, Facebook!");
        sm2.postMessage("Hello, Twitter!");
    }
}
```

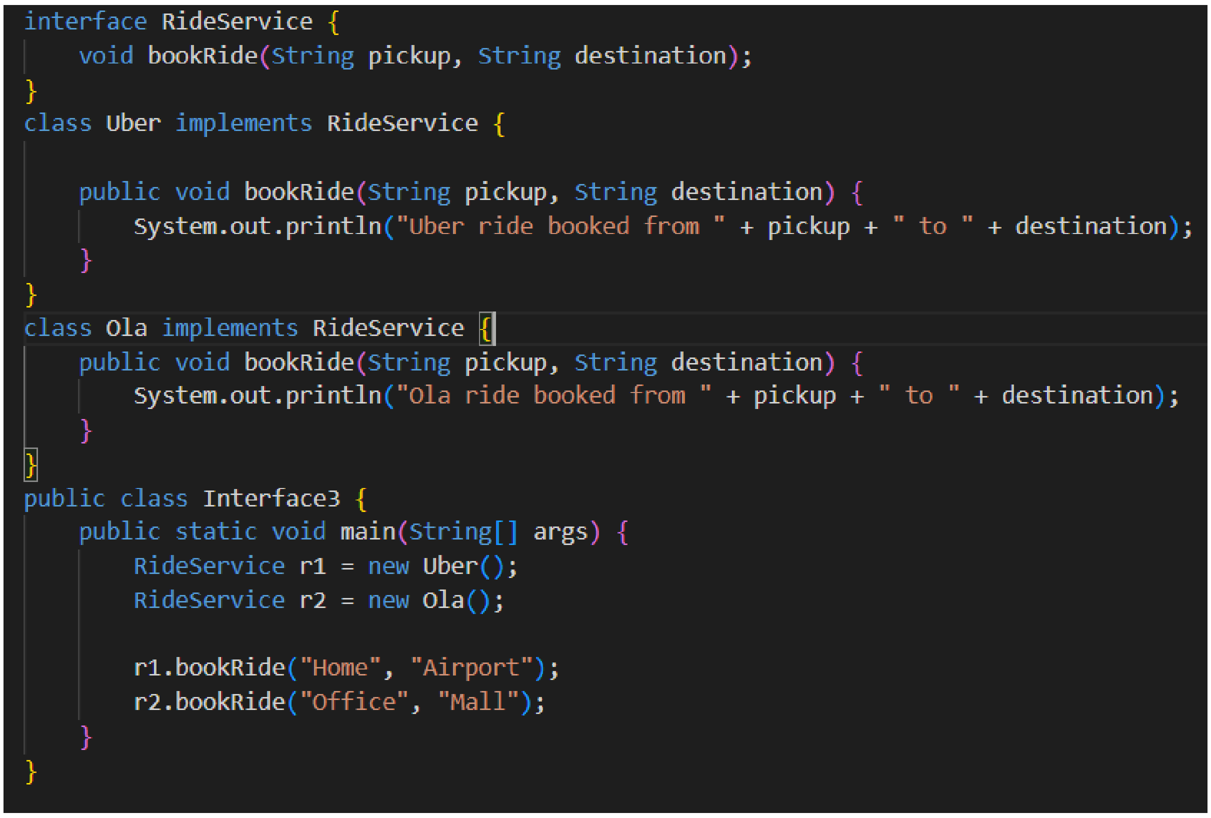
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac Interface2.java

D:\oops>java Interface2
Posting on Facebook: Hello, Facebook!
Tweeting on Twitter: Hello, Twitter!
```

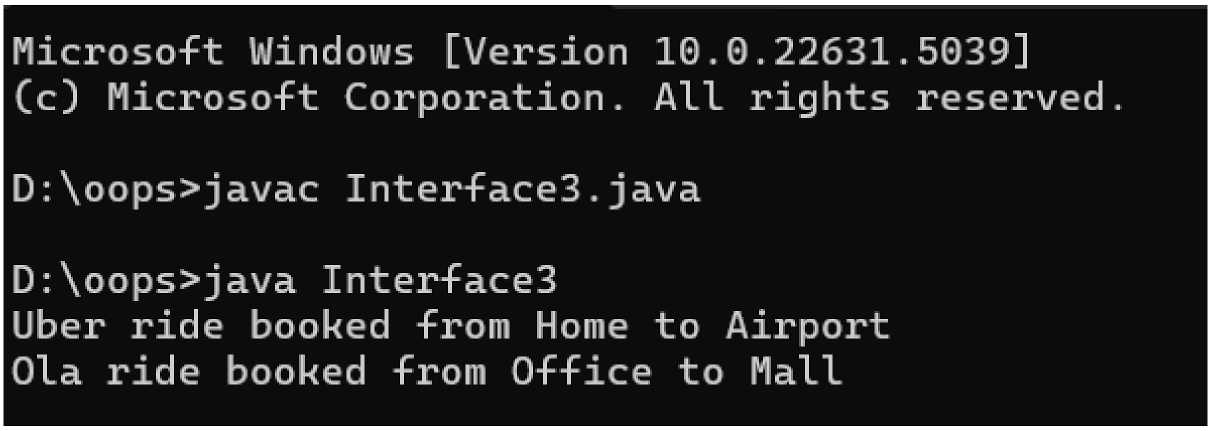
12c) Taxi Service

Code:



```java
interface RideService {
    void bookRide(String pickup, String destination);
}
class Uber implements RideService {

    public void bookRide(String pickup, String destination) {
        System.out.println("Uber ride booked from " + pickup + " to " + destination);
    }
}
class Ola implements RideService {
    public void bookRide(String pickup, String destination) {
        System.out.println("Ola ride booked from " + pickup + " to " + destination);
    }
}
public class Interface3 {
    public static void main(String[] args) {
        RideService r1 = new Uber();
        RideService r2 = new Ola();

        r1.bookRide("Home", "Airport");
        r2.bookRide("Office", "Mall");
    }
}
```

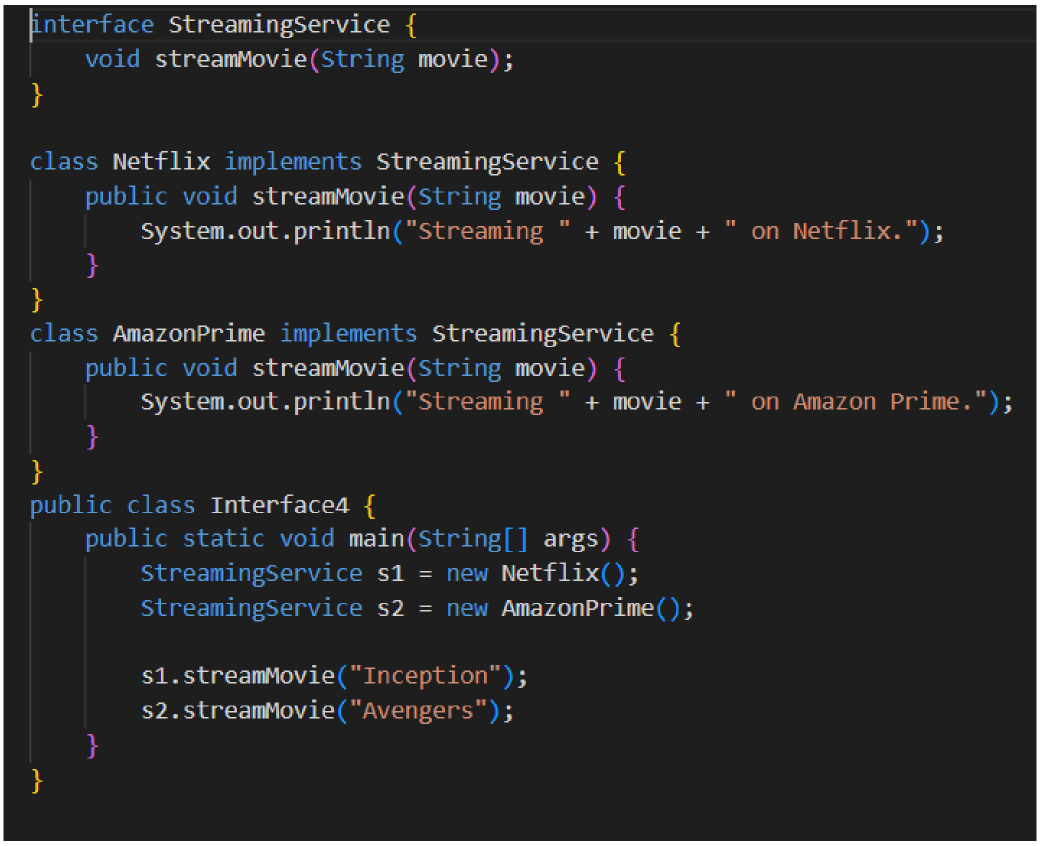
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac Interface3.java

D:\oops>java Interface3
Uber ride booked from Home to Airport
Ola ride booked from Office to Mall
```

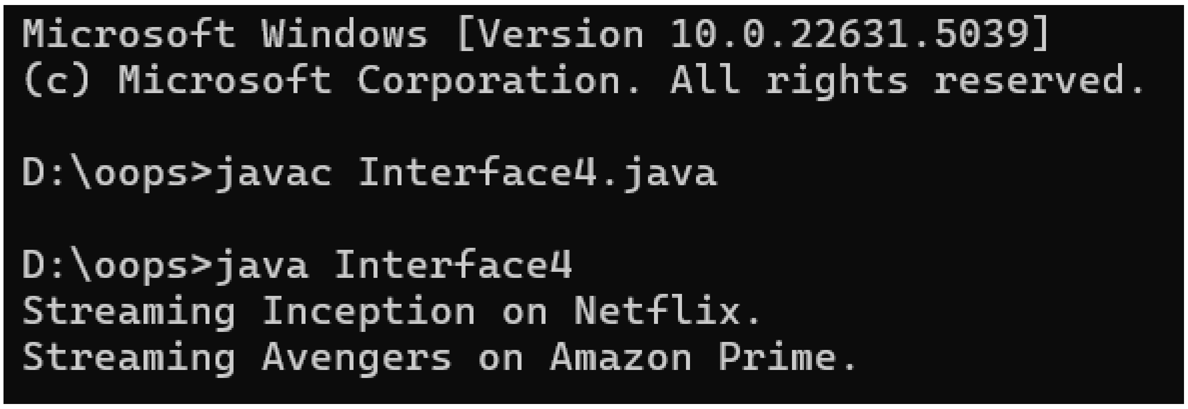
12d) Streaming Service

Code:



```java
interface StreamingService {
    void streamMovie(String movie);
}

class Netflix implements StreamingService {
    public void streamMovie(String movie) {
        System.out.println("Streaming " + movie + " on Netflix.");
    }
}
class AmazonPrime implements StreamingService {
    public void streamMovie(String movie) {
        System.out.println("Streaming " + movie + " on Amazon Prime.");
    }
}
public class Interface4 {
    public static void main(String[] args) {
        StreamingService s1 = new Netflix();
        StreamingService s2 = new AmazonPrime();

        s1.streamMovie("Inception");
        s2.streamMovie("Avengers");
    }
}
```

Output:

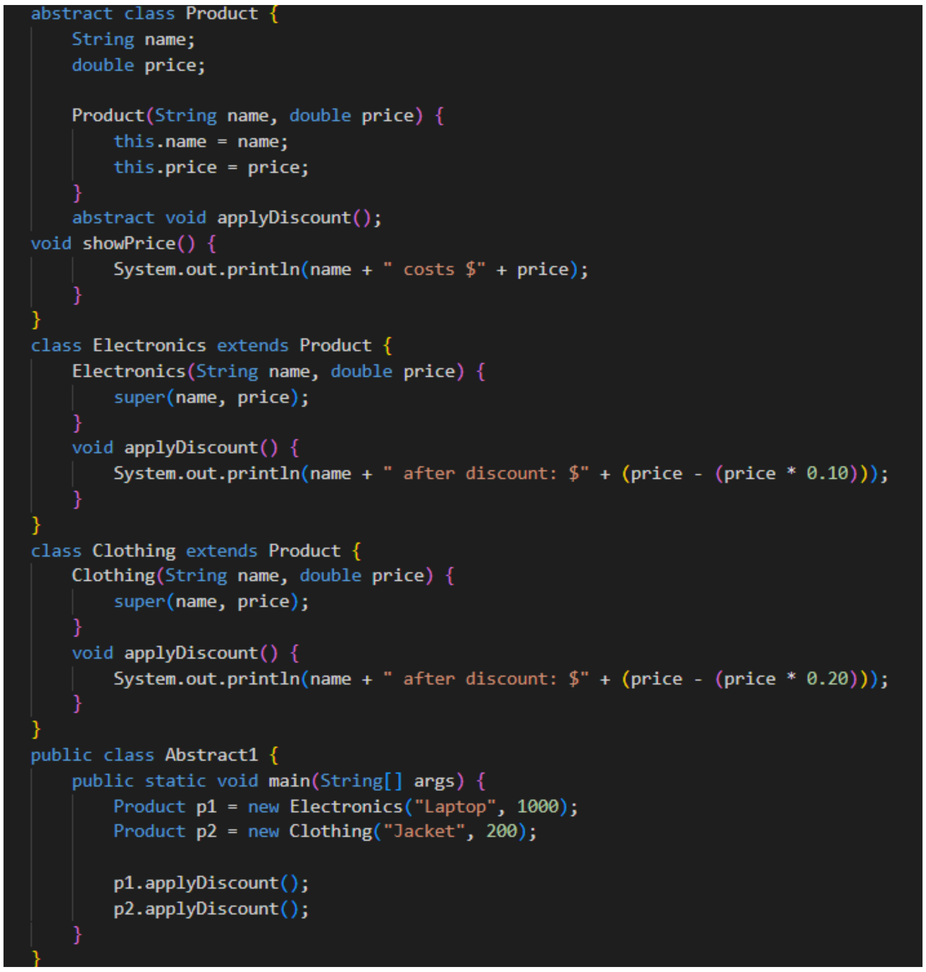


```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac Interface4.java

D:\oops>java Interface4
Streaming Inception on Netflix.
Streaming Avengers on Amazon Prime.
```

13) ABSTRACT CLASS PROGRAMS

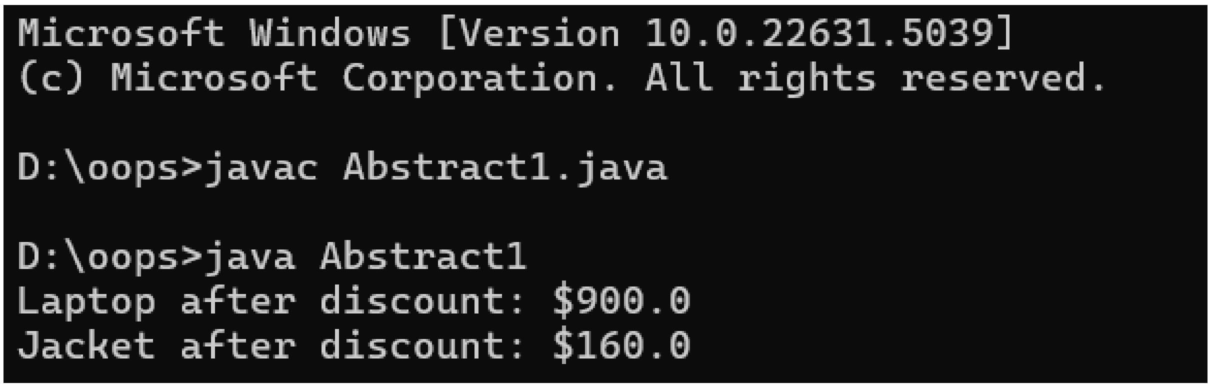
13 a) Online Shopping

Code:



```
abstract class Product {
    String name;
    double price;

    Product(String name, double price) {
        this.name = name;
        this.price = price;
    }
    abstract void applyDiscount();
void showPrice() {
        System.out.println(name + " costs $" + price);
    }
}
class Electronics extends Product {
    Electronics(String name, double price) {
        super(name, price);
    }
    void applyDiscount() {
        System.out.println(name + " after discount: $" + (price - (price * 0.10)));
    }
}
class Clothing extends Product {
    Clothing(String name, double price) {
        super(name, price);
    }
    void applyDiscount() {
        System.out.println(name + " after discount: $" + (price - (price * 0.20)));
    }
}
public class Abstract1 {
    public static void main(String[] args) {
        Product p1 = new Electronics("Laptop", 1000);
        Product p2 = new Clothing("Jacket", 200);

        p1.applyDiscount();
        p2.applyDiscount();
    }
}
```

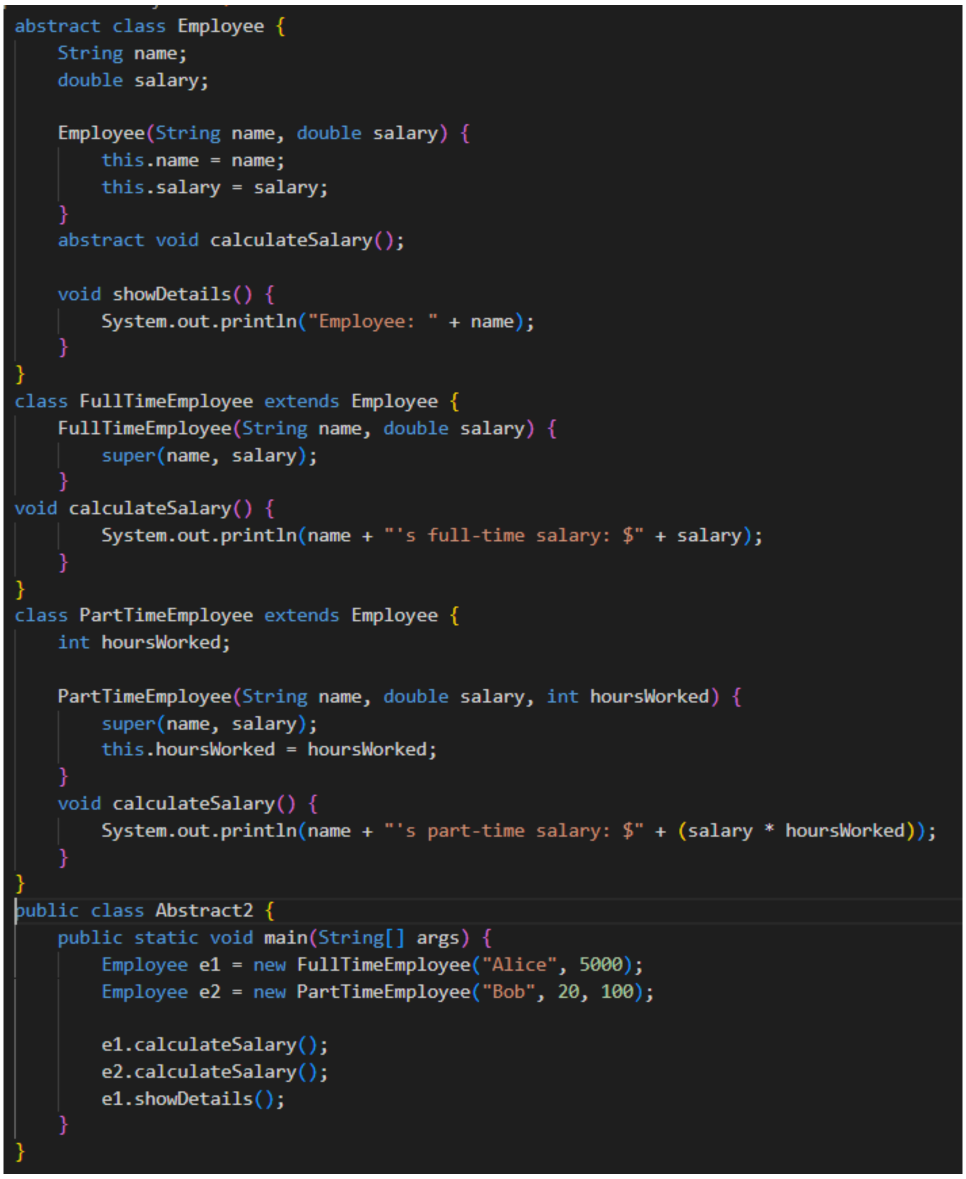
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac Abstract1.java

D:\oops>java Abstract1
Laptop after discount: $900.0
Jacket after discount: $160.0
```

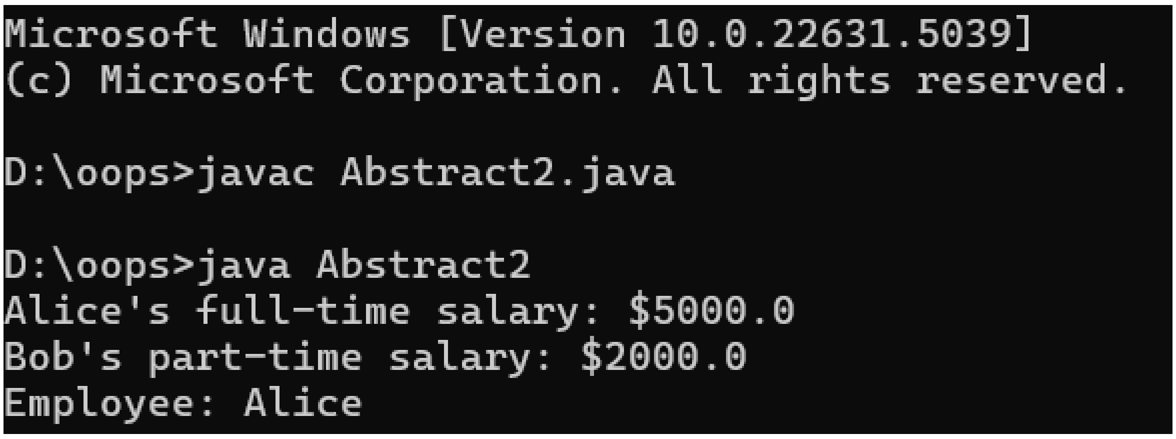
13 b) Employee Salary Calculation

Code:



```java
abstract class Employee {
    String name;
    double salary;

    Employee(String name, double salary) {
        this.name = name;
        this.salary = salary;
    }
    abstract void calculateSalary();

    void showDetails() {
        System.out.println("Employee: " + name);
    }
}
class FullTimeEmployee extends Employee {
    FullTimeEmployee(String name, double salary) {
        super(name, salary);
    }
void calculateSalary() {
        System.out.println(name + "'s full-time salary: $" + salary);
    }
}
class PartTimeEmployee extends Employee {
    int hoursWorked;

    PartTimeEmployee(String name, double salary, int hoursWorked) {
        super(name, salary);
        this.hoursWorked = hoursWorked;
    }
    void calculateSalary() {
        System.out.println(name + "'s part-time salary: $" + (salary * hoursWorked));
    }
}
public class Abstract2 {
    public static void main(String[] args) {
        Employee e1 = new FullTimeEmployee("Alice", 5000);
        Employee e2 = new PartTimeEmployee("Bob", 20, 100);

        e1.calculateSalary();
        e2.calculateSalary();
        e1.showDetails();
    }
}
```

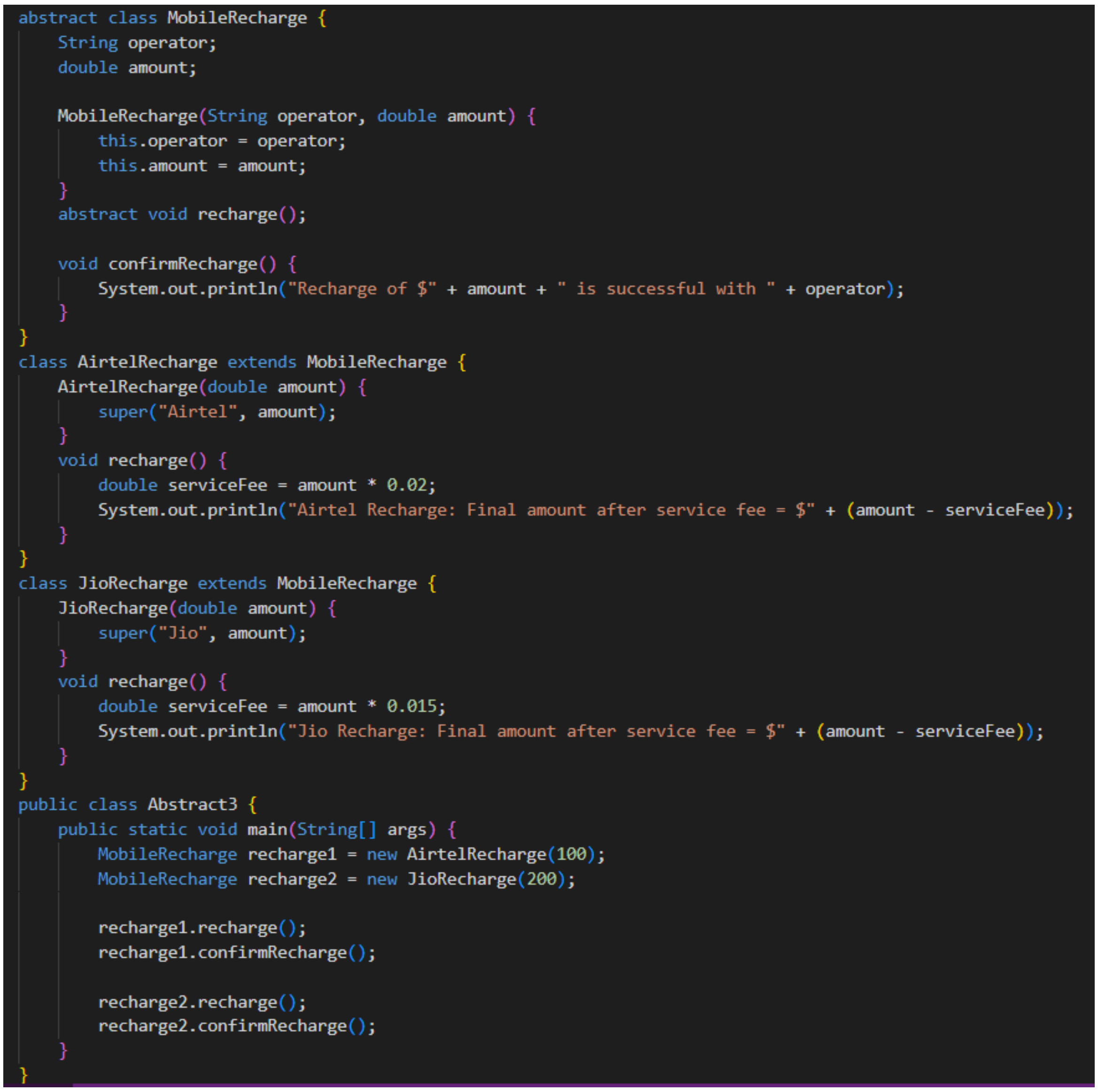
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac Abstract2.java

D:\oops>java Abstract2
Alice's full-time salary: $5000.0
Bob's part-time salary: $2000.0
Employee: Alice
```

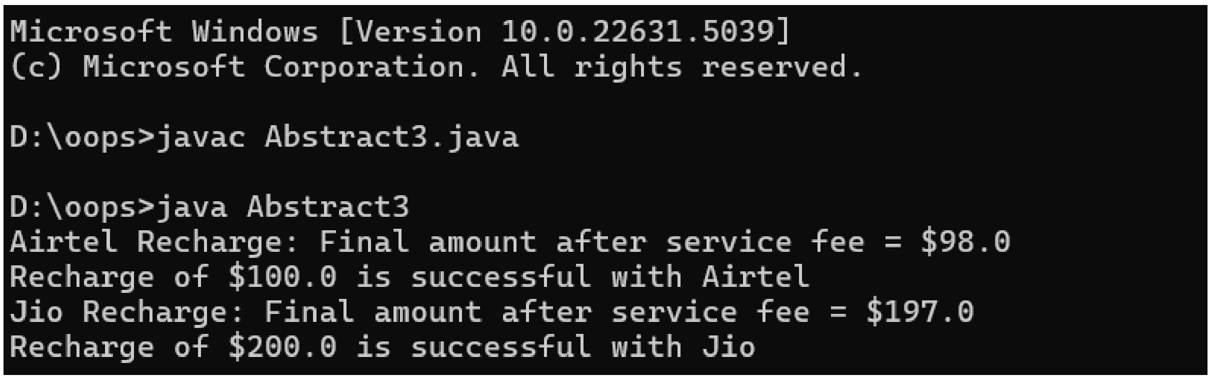
13 c) Mobile Recharge

Code:



```java
abstract class MobileRecharge {
    String operator;
    double amount;

    MobileRecharge(String operator, double amount) {
        this.operator = operator;
        this.amount = amount;
    }
    abstract void recharge();

    void confirmRecharge() {
        System.out.println("Recharge of $" + amount + " is successful with " + operator);
    }
}
class AirtelRecharge extends MobileRecharge {
    AirtelRecharge(double amount) {
        super("Airtel", amount);
    }
    void recharge() {
        double serviceFee = amount * 0.02;
        System.out.println("Airtel Recharge: Final amount after service fee = $" + (amount - serviceFee));
    }
}
class JioRecharge extends MobileRecharge {
    JioRecharge(double amount) {
        super("Jio", amount);
    }
    void recharge() {
        double serviceFee = amount * 0.015;
        System.out.println("Jio Recharge: Final amount after service fee = $" + (amount - serviceFee));
    }
}
public class Abstract3 {
    public static void main(String[] args) {
        MobileRecharge recharge1 = new AirtelRecharge(100);
        MobileRecharge recharge2 = new JioRecharge(200);

        recharge1.recharge();
        recharge1.confirmRecharge();

        recharge2.recharge();
        recharge2.confirmRecharge();
    }
}
```

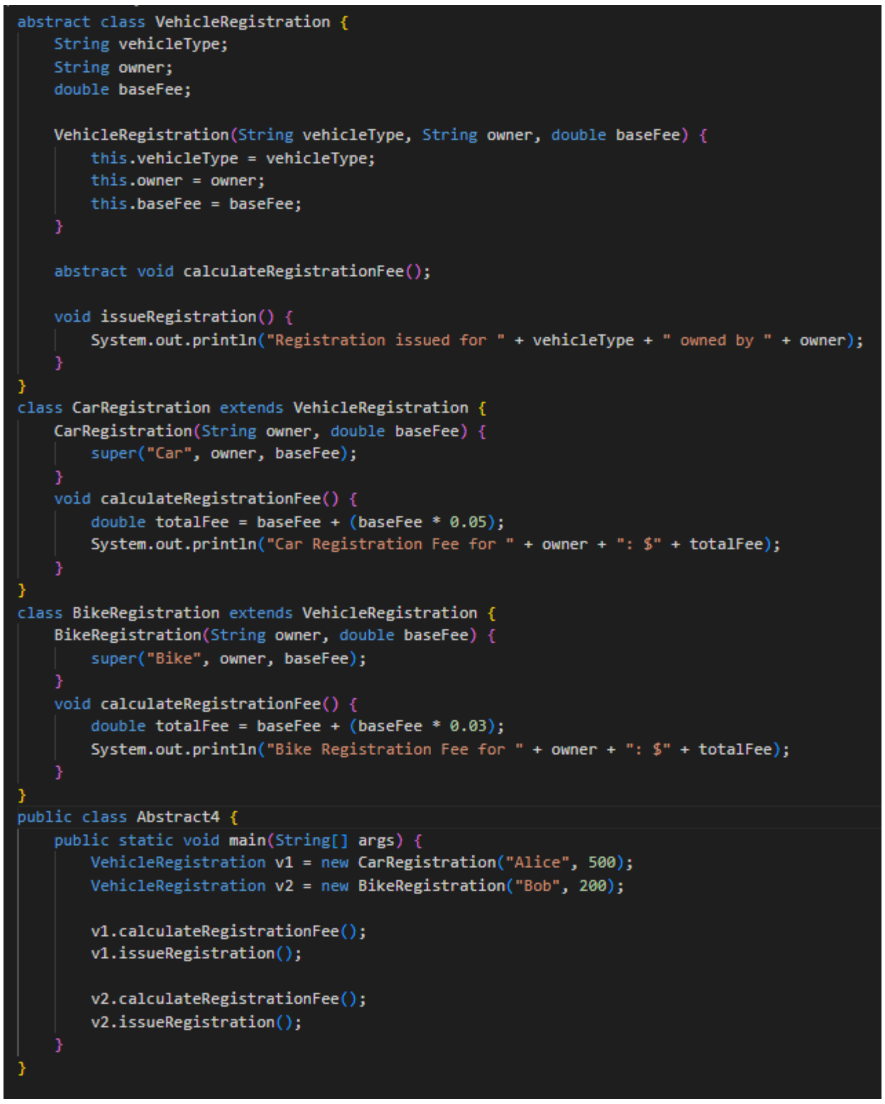
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac Abstract3.java

D:\oops>java Abstract3
Airtel Recharge: Final amount after service fee = $98.0
Recharge of $100.0 is successful with Airtel
Jio Recharge: Final amount after service fee = $197.0
Recharge of $200.0 is successful with Jio
```

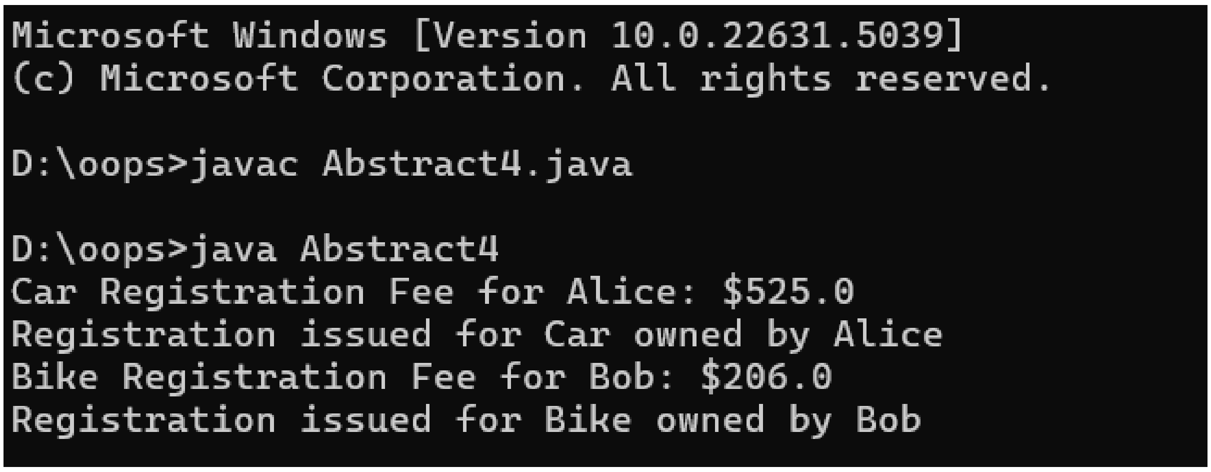
13 d) Vehicle Registration

Code:



```
abstract class VehicleRegistration {
    String vehicleType;
    String owner;
    double baseFee;

    VehicleRegistration(String vehicleType, String owner, double baseFee) {
        this.vehicleType = vehicleType;
        this.owner = owner;
        this.baseFee = baseFee;
    }

    abstract void calculateRegistrationFee();

    void issueRegistration() {
        System.out.println("Registration issued for " + vehicleType + " owned by " + owner);
    }
}
class CarRegistration extends VehicleRegistration {
    CarRegistration(String owner, double baseFee) {
        super("Car", owner, baseFee);
    }
    void calculateRegistrationFee() {
        double totalFee = baseFee + (baseFee * 0.05);
        System.out.println("Car Registration Fee for " + owner + ": $" + totalFee);
    }
}
class BikeRegistration extends VehicleRegistration {
    BikeRegistration(String owner, double baseFee) {
        super("Bike", owner, baseFee);
    }
    void calculateRegistrationFee() {
        double totalFee = baseFee + (baseFee * 0.03);
        System.out.println("Bike Registration Fee for " + owner + ": $" + totalFee);
    }
}
public class Abstract4 {
    public static void main(String[] args) {
        VehicleRegistration v1 = new CarRegistration("Alice", 500);
        VehicleRegistration v2 = new BikeRegistration("Bob", 200);

        v1.calculateRegistrationFee();
        v1.issueRegistration();

        v2.calculateRegistrationFee();
        v2.issueRegistration();
    }
}
```

Output:



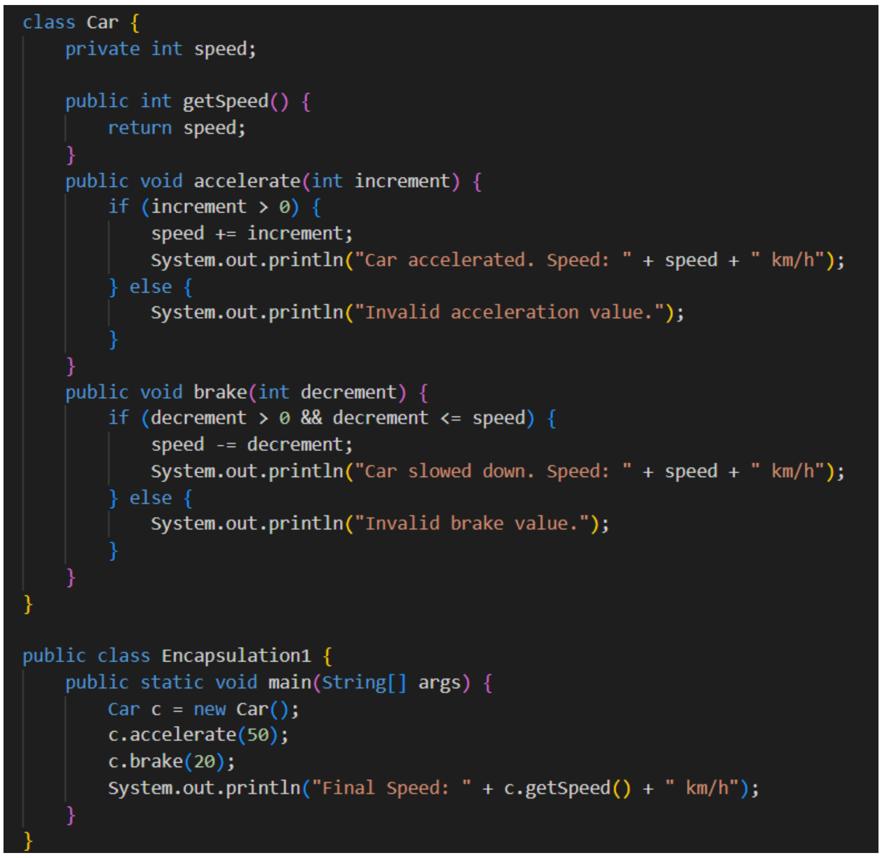
```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac Abstract4.java

D:\oops>java Abstract4
Car Registration Fee for Alice: $525.0
Registration issued for Car owned by Alice
Bike Registration Fee for Bob: $206.0
Registration issued for Bike owned by Bob
```

14)ENCAPSULATION

ENCAPSULATION PROGRAMS

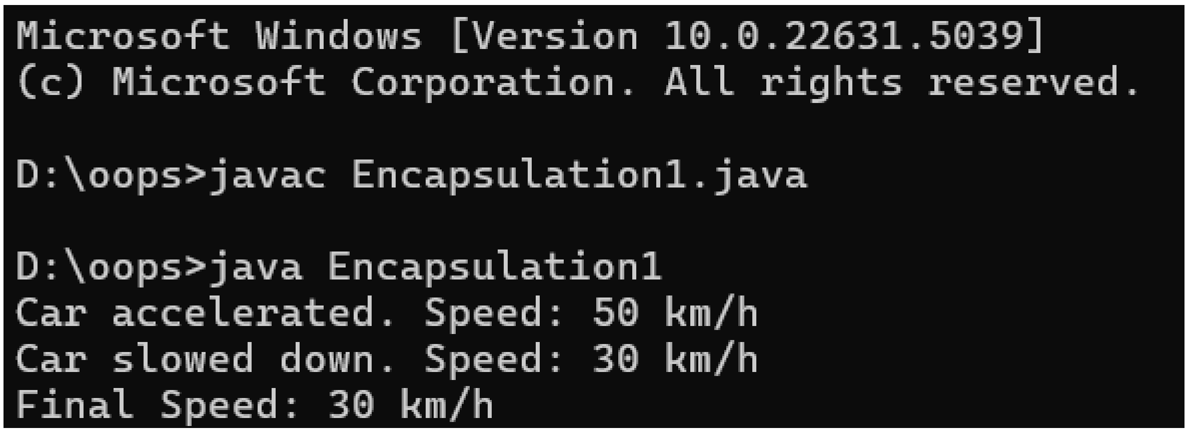
14a) Car Speed Control System

Code:



```java
class Car {
    private int speed;

    public int getSpeed() {
        return speed;
    }
    public void accelerate(int increment) {
        if (increment > 0) {
            speed += increment;
            System.out.println("Car accelerated. Speed: " + speed + " km/h");
        } else {
            System.out.println("Invalid acceleration value.");
        }
    }
    public void brake(int decrement) {
        if (decrement > 0 && decrement <= speed) {
            speed -= decrement;
            System.out.println("Car slowed down. Speed: " + speed + " km/h");
        } else {
            System.out.println("Invalid brake value.");
        }
    }
}

public class Encapsulation1 {
    public static void main(String[] args) {
        Car c = new Car();
        c.accelerate(50);
        c.brake(20);
        System.out.println("Final Speed: " + c.getSpeed() + " km/h");
    }
}
```

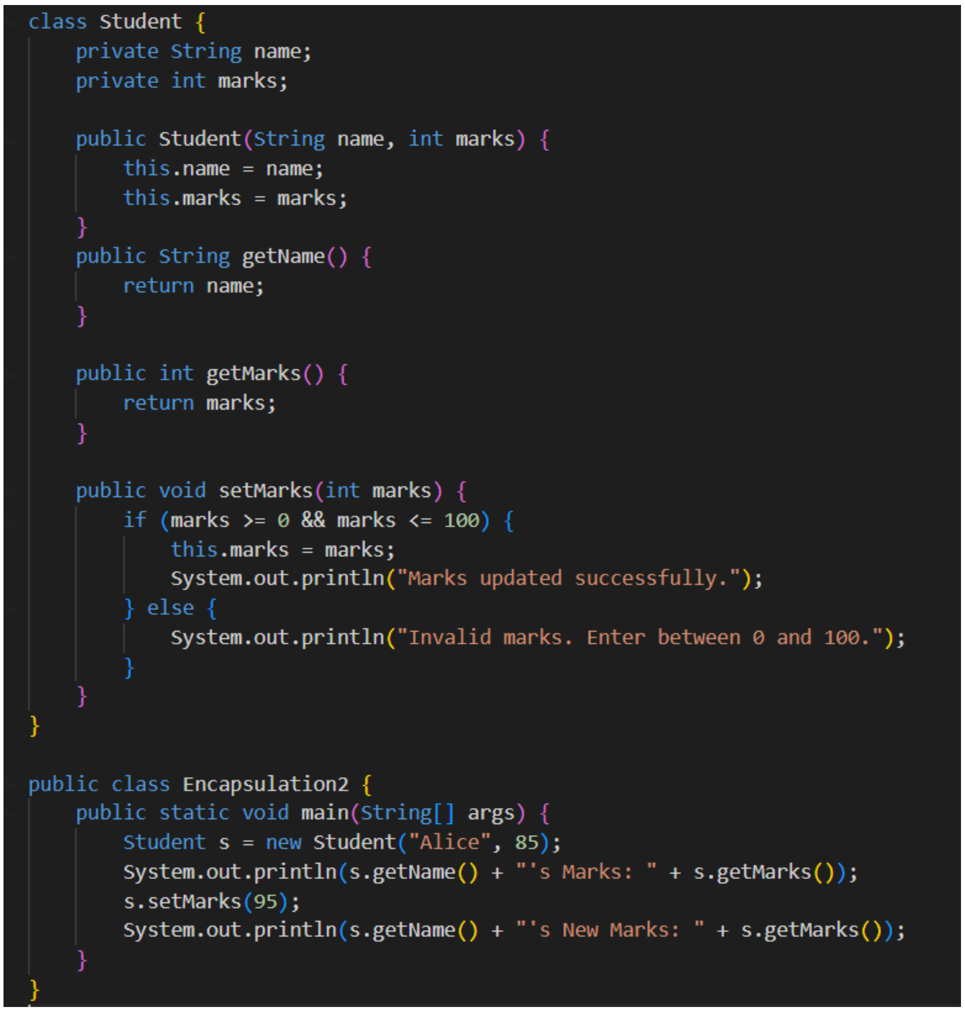
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac Encapsulation1.java

D:\oops>java Encapsulation1
Car accelerated. Speed: 50 km/h
Car slowed down. Speed: 30 km/h
Final Speed: 30 km/h
```

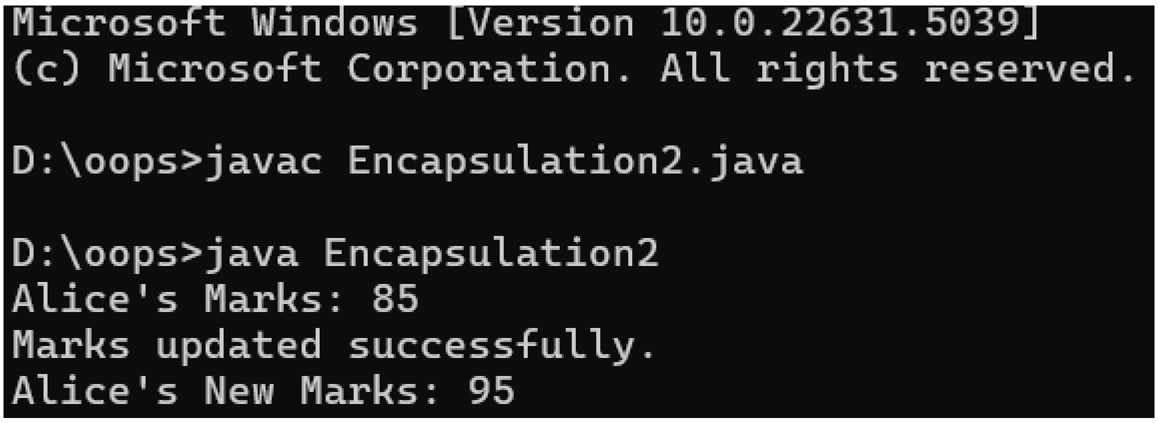
14b) Student Management System

Code:



```java
class Student {
    private String name;
    private int marks;

    public Student(String name, int marks) {
        this.name = name;
        this.marks = marks;
    }
    public String getName() {
        return name;
    }

    public int getMarks() {
        return marks;
    }

    public void setMarks(int marks) {
        if (marks >= 0 && marks <= 100) {
            this.marks = marks;
            System.out.println("Marks updated successfully.");
        } else {
            System.out.println("Invalid marks. Enter between 0 and 100.");
        }
    }
}

public class Encapsulation2 {
    public static void main(String[] args) {
        Student s = new Student("Alice", 85);
        System.out.println(s.getName() + "'s Marks: " + s.getMarks());
        s.setMarks(95);
        System.out.println(s.getName() + "'s New Marks: " + s.getMarks());
    }
}
```

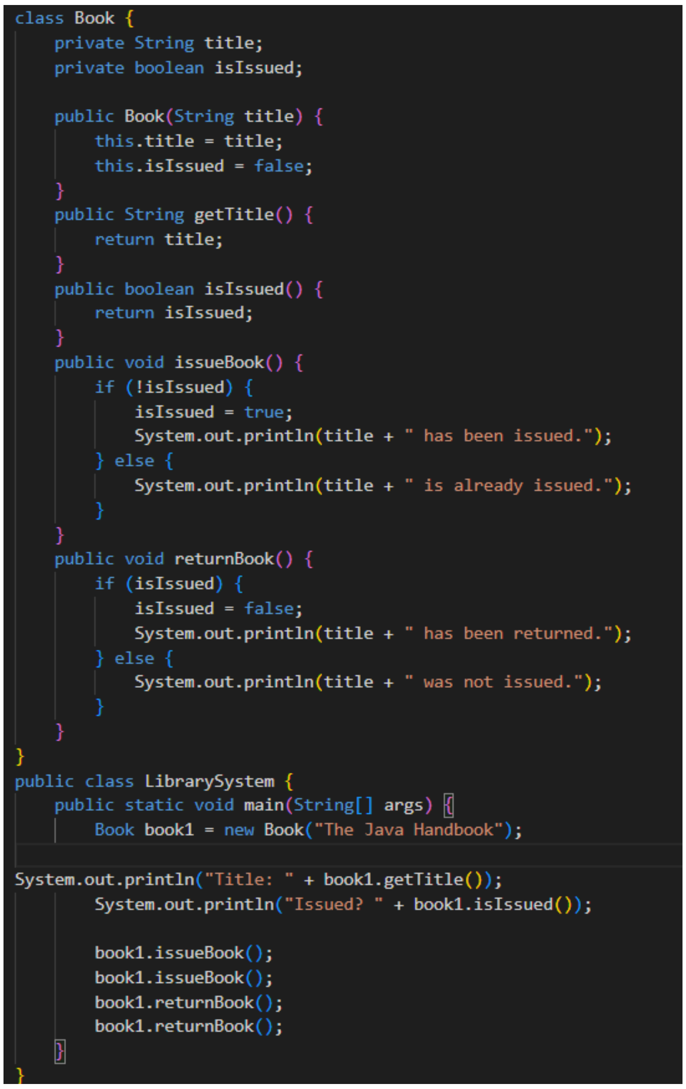
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac Encapsulation2.java

D:\oops>java Encapsulation2
Alice's Marks: 85
Marks updated successfully.
Alice's New Marks: 95
```

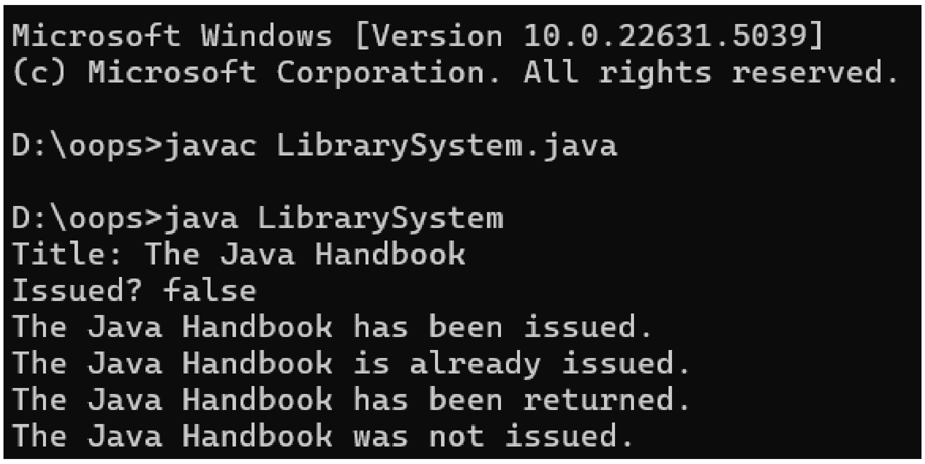
14c) Library Management System

Code:



```java
class Book {
    private String title;
    private boolean isIssued;

    public Book(String title) {
        this.title = title;
        this.isIssued = false;
    }
    public String getTitle() {
        return title;
    }
    public boolean isIssued() {
        return isIssued;
    }
    public void issueBook() {
        if (!isIssued) {
            isIssued = true;
            System.out.println(title + " has been issued.");
        } else {
            System.out.println(title + " is already issued.");
        }
    }
    public void returnBook() {
        if (isIssued) {
            isIssued = false;
            System.out.println(title + " has been returned.");
        } else {
            System.out.println(title + " was not issued.");
        }
    }
}
public class LibrarySystem {
    public static void main(String[] args) {
        Book book1 = new Book("The Java Handbook");

System.out.println("Title: " + book1.getTitle());
        System.out.println("Issued? " + book1.isIssued());

        book1.issueBook();
        book1.issueBook();
        book1.returnBook();
        book1.returnBook();
    }
}
```

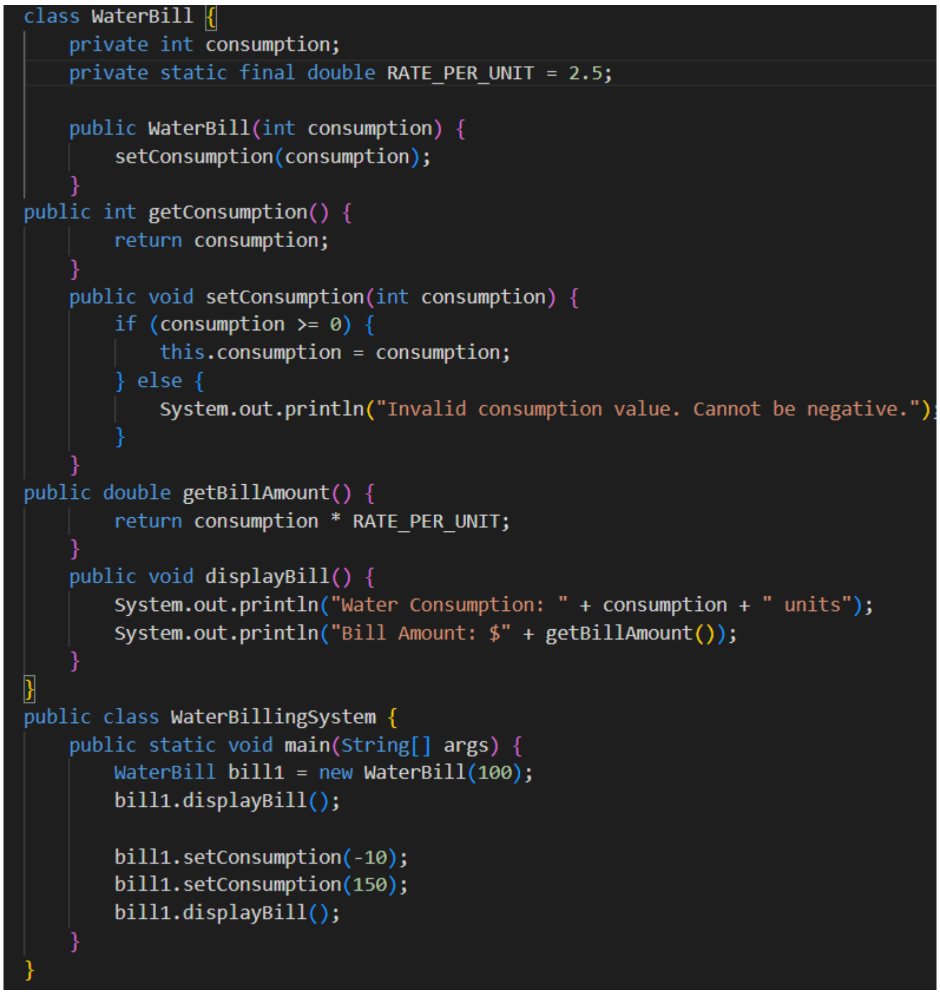
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac LibrarySystem.java

D:\oops>java LibrarySystem
Title: The Java Handbook
Issued? false
The Java Handbook has been issued.
The Java Handbook is already issued.
The Java Handbook has been returned.
The Java Handbook was not issued.
```

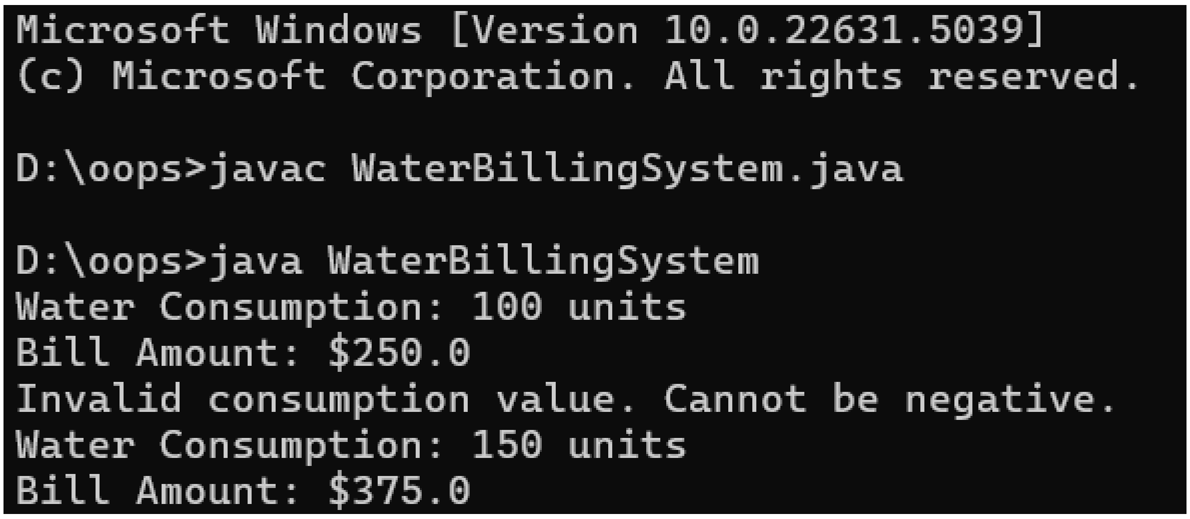
14d) Water billingSystem

Code:



```
class WaterBill {
    private int consumption;
    private static final double RATE_PER_UNIT = 2.5;

    public WaterBill(int consumption) {
        setConsumption(consumption);
    }
public int getConsumption() {
        return consumption;
    }
    public void setConsumption(int consumption) {
        if (consumption >= 0) {
            this.consumption = consumption;
        } else {
            System.out.println("Invalid consumption value. Cannot be negative.")
        }
    }
public double getBillAmount() {
        return consumption * RATE_PER_UNIT;
    }
    public void displayBill() {
        System.out.println("Water Consumption: " + consumption + " units");
        System.out.println("Bill Amount: $" + getBillAmount());
    }
}
public class WaterBillingSystem {
    public static void main(String[] args) {
        WaterBill bill1 = new WaterBill(100);
        bill1.displayBill();

        bill1.setConsumption(-10);
        bill1.setConsumption(150);
        bill1.displayBill();
    }
}
```

Output:

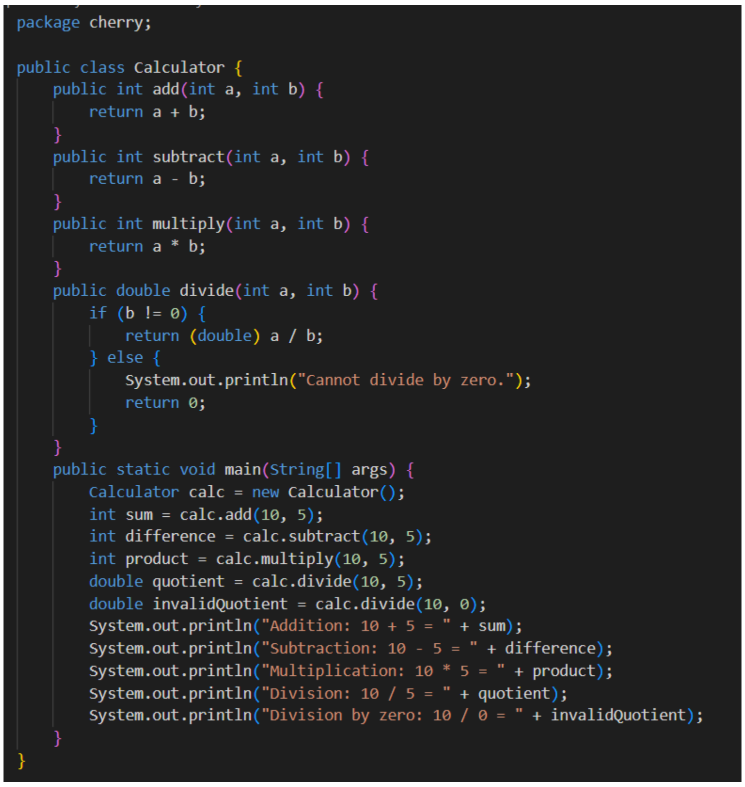


```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac WaterBillingSystem.java

D:\oops>java WaterBillingSystem
Water Consumption: 100 units
Bill Amount: $250.0
Invalid consumption value. Cannot be negative.
Water Consumption: 150 units
Bill Amount: $375.0
```

15)PACKAGES PROGRAMS

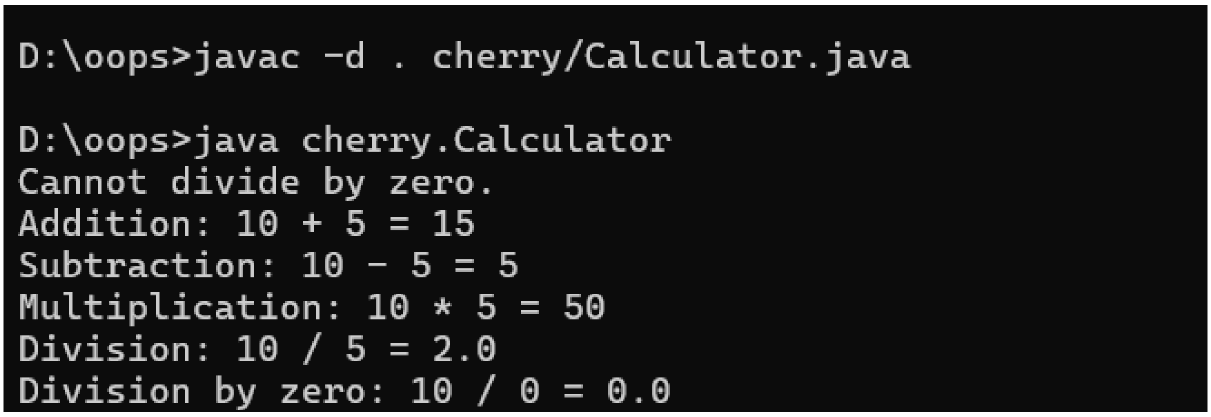
15a) User Defined Packages

Code:



```java
package cherry;

public class Calculator {
    public int add(int a, int b) {
        return a + b;
    }
    public int subtract(int a, int b) {
        return a - b;
    }
    public int multiply(int a, int b) {
        return a * b;
    }
    public double divide(int a, int b) {
        if (b != 0) {
            return (double) a / b;
        } else {
            System.out.println("Cannot divide by zero.");
            return 0;
        }
    }
    public static void main(String[] args) {
        Calculator calc = new Calculator();
        int sum = calc.add(10, 5);
        int difference = calc.subtract(10, 5);
        int product = calc.multiply(10, 5);
        double quotient = calc.divide(10, 5);
        double invalidQuotient = calc.divide(10, 0);
        System.out.println("Addition: 10 + 5 = " + sum);
        System.out.println("Subtraction: 10 - 5 = " + difference);
        System.out.println("Multiplication: 10 * 5 = " + product);
        System.out.println("Division: 10 / 5 = " + quotient);
        System.out.println("Division by zero: 10 / 0 = " + invalidQuotient);
    }
}
```

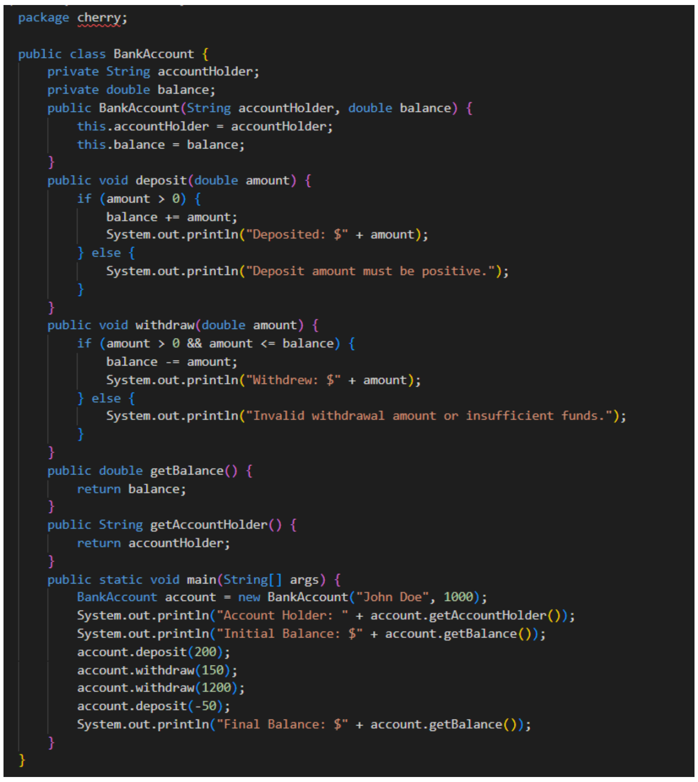
Output:



```
D:\oops>javac -d . cherry/Calculator.java

D:\oops>java cherry.Calculator
Cannot divide by zero.
Addition: 10 + 5 = 15
Subtraction: 10 - 5 = 5
Multiplication: 10 * 5 = 50
Division: 10 / 5 = 2.0
Division by zero: 10 / 0 = 0.0
```

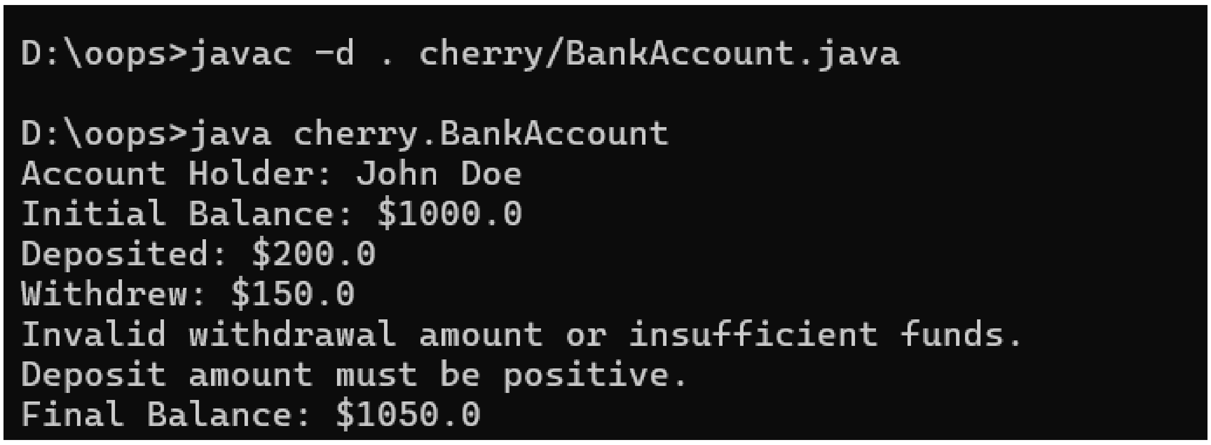
15b) User Defined Packages

Code:



```java
package cherry;

public class BankAccount {
    private String accountHolder;
    private double balance;
    public BankAccount(String accountHolder, double balance) {
        this.accountHolder = accountHolder;
        this.balance = balance;
    }
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: $" + amount);
        } else {
            System.out.println("Deposit amount must be positive.");
        }
    }
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew: $" + amount);
        } else {
            System.out.println("Invalid withdrawal amount or insufficient funds.");
        }
    }
    public double getBalance() {
        return balance;
    }
    public String getAccountHolder() {
        return accountHolder;
    }
    public static void main(String[] args) {
        BankAccount account = new BankAccount("John Doe", 1000);
        System.out.println("Account Holder: " + account.getAccountHolder());
        System.out.println("Initial Balance: $" + account.getBalance());
        account.deposit(200);
        account.withdraw(150);
        account.withdraw(1200);
        account.deposit(-50);
        System.out.println("Final Balance: $" + account.getBalance());
    }
}
```

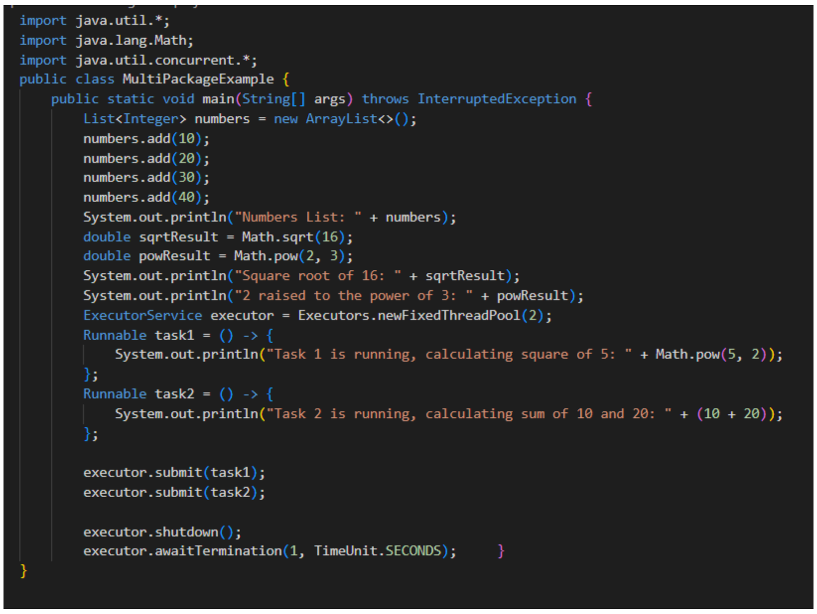
Output:



```
D:\oops>javac -d . cherry/BankAccount.java

D:\oops>java cherry.BankAccount
Account Holder: John Doe
Initial Balance: $1000.0
Deposited: $200.0
Withdrew: $150.0
Invalid withdrawal amount or insufficient funds.
Deposit amount must be positive.
Final Balance: $1050.0
```

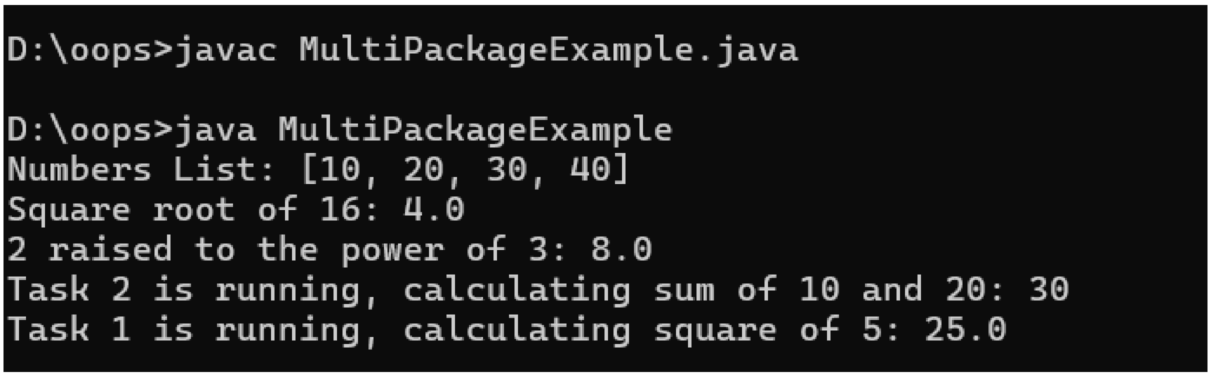
15c) Built – in Package (3 Packages)

Code:



```java
import java.util.*;
import java.lang.Math;
import java.util.concurrent.*;
public class MultiPackageExample {
    public static void main(String[] args) throws InterruptedException {
        List<Integer> numbers = new ArrayList<>();
        numbers.add(10);
        numbers.add(20);
        numbers.add(30);
        numbers.add(40);
        System.out.println("Numbers List: " + numbers);
        double sqrtResult = Math.sqrt(16);
        double powResult = Math.pow(2, 3);
        System.out.println("Square root of 16: " + sqrtResult);
        System.out.println("2 raised to the power of 3: " + powResult);
        ExecutorService executor = Executors.newFixedThreadPool(2);
        Runnable task1 = () -> {
            System.out.println("Task 1 is running, calculating square of 5: " + Math.pow(5, 2));
        };
        Runnable task2 = () -> {
            System.out.println("Task 2 is running, calculating sum of 10 and 20: " + (10 + 20));
        };

        executor.submit(task1);
        executor.submit(task2);

        executor.shutdown();
        executor.awaitTermination(1, TimeUnit.SECONDS);     }
}
```

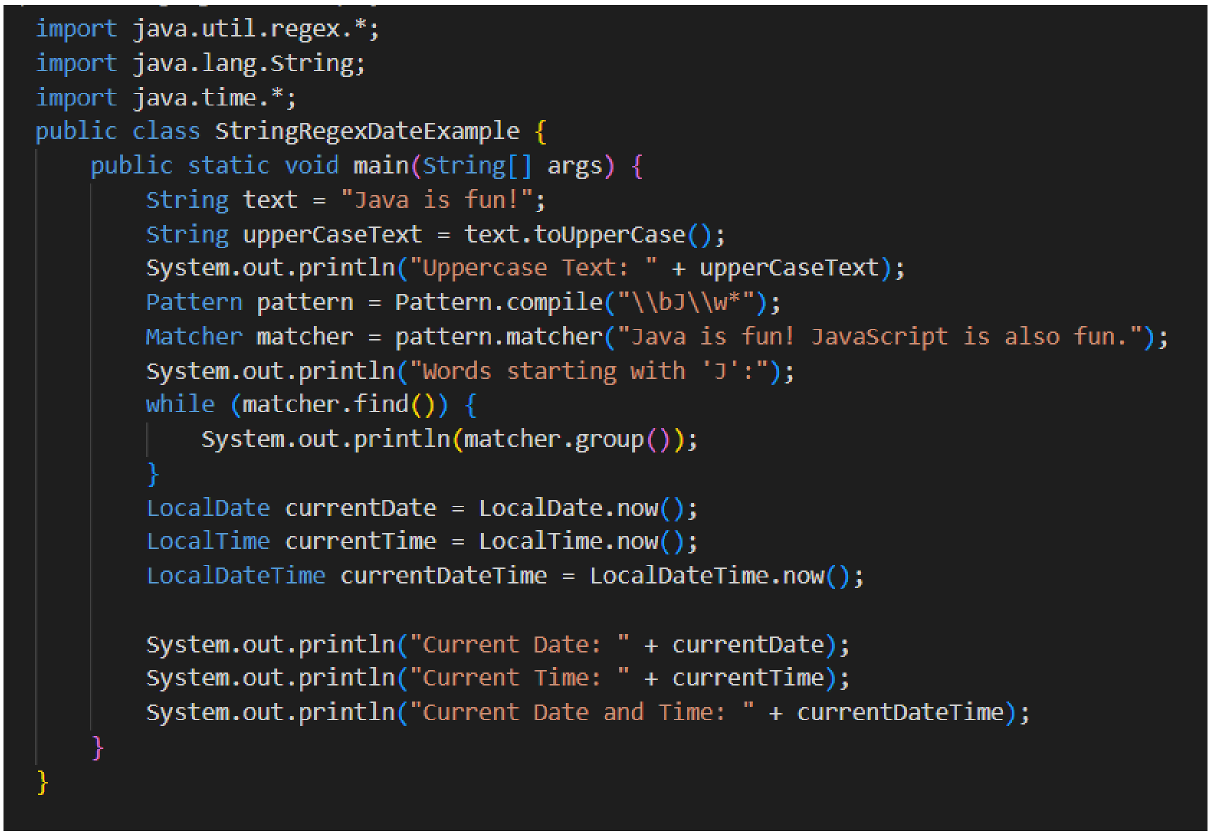
Output:



```
D:\oops>javac MultiPackageExample.java

D:\oops>java MultiPackageExample
Numbers List: [10, 20, 30, 40]
Square root of 16: 4.0
2 raised to the power of 3: 8.0
Task 2 is running, calculating sum of 10 and 20: 30
Task 1 is running, calculating square of 5: 25.0
```

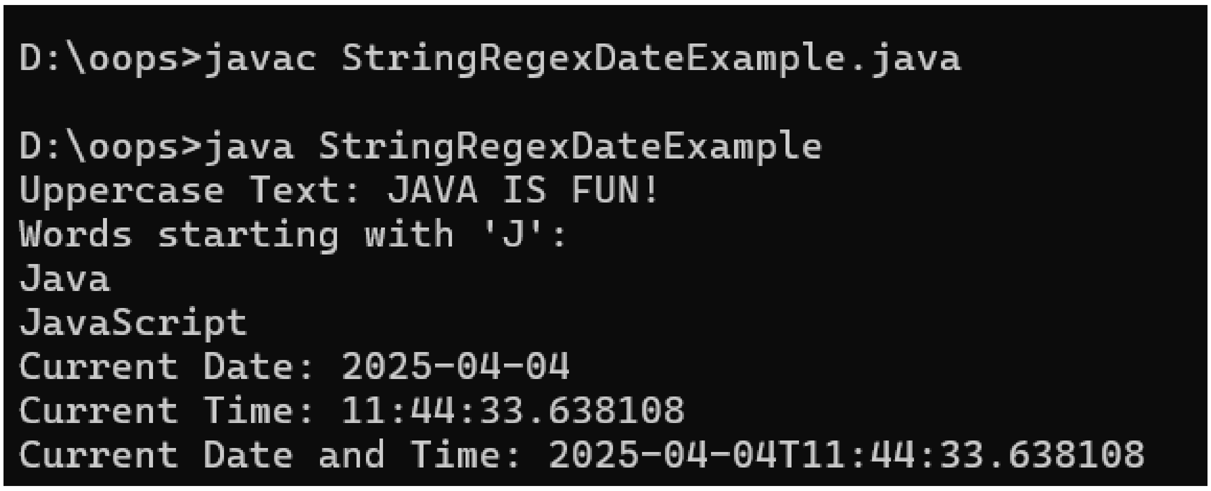
15d) Built – in Package (3 Packages)

Code:



```java
import java.util.regex.*;
import java.lang.String;
import java.time.*;
public class StringRegexDateExample {
    public static void main(String[] args) {
        String text = "Java is fun!";
        String upperCaseText = text.toUpperCase();
        System.out.println("Uppercase Text: " + upperCaseText);
        Pattern pattern = Pattern.compile("\\bJ\\w*");
        Matcher matcher = pattern.matcher("Java is fun! JavaScript is also fun.");
        System.out.println("Words starting with 'J':");
        while (matcher.find()) {
            System.out.println(matcher.group());
        }
        LocalDate currentDate = LocalDate.now();
        LocalTime currentTime = LocalTime.now();
        LocalDateTime currentDateTime = LocalDateTime.now();

        System.out.println("Current Date: " + currentDate);
        System.out.println("Current Time: " + currentTime);
        System.out.println("Current Date and Time: " + currentDateTime);
    }
}
```

Output:

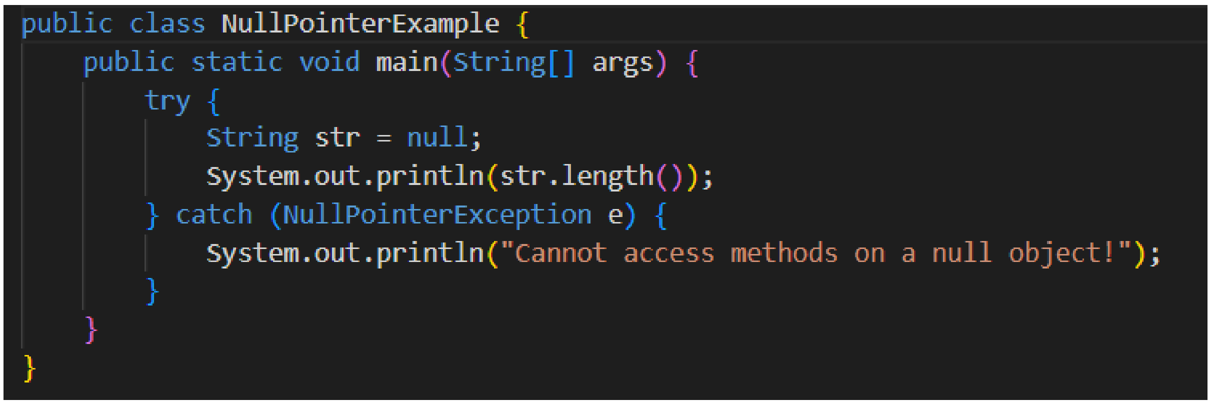


```
D:\oops>javac StringRegexDateExample.java

D:\oops>java StringRegexDateExample
Uppercase Text: JAVA IS FUN!
Words starting with 'J':
Java
JavaScript
Current Date: 2025-04-04
Current Time: 11:44:33.638108
Current Date and Time: 2025-04-04T11:44:33.638108
```

16)EXCEPTION HANDLING PROGRAMS

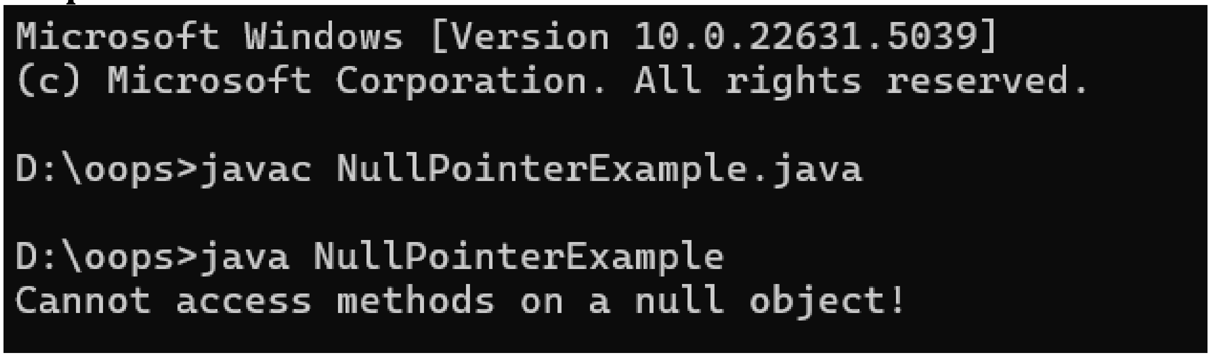
16a)NullPointerExample

Code:



```java
public class NullPointerExample {
    public static void main(String[] args) {
        try {
            String str = null;
            System.out.println(str.length());
        } catch (NullPointerException e) {
            System.out.println("Cannot access methods on a null object!");
        }
    }
}
```

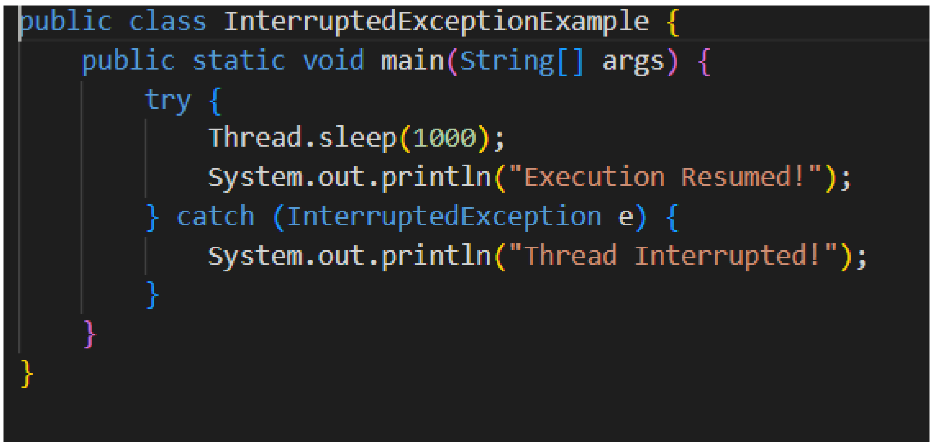
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac NullPointerExample.java

D:\oops>java NullPointerExample
Cannot access methods on a null object!
```

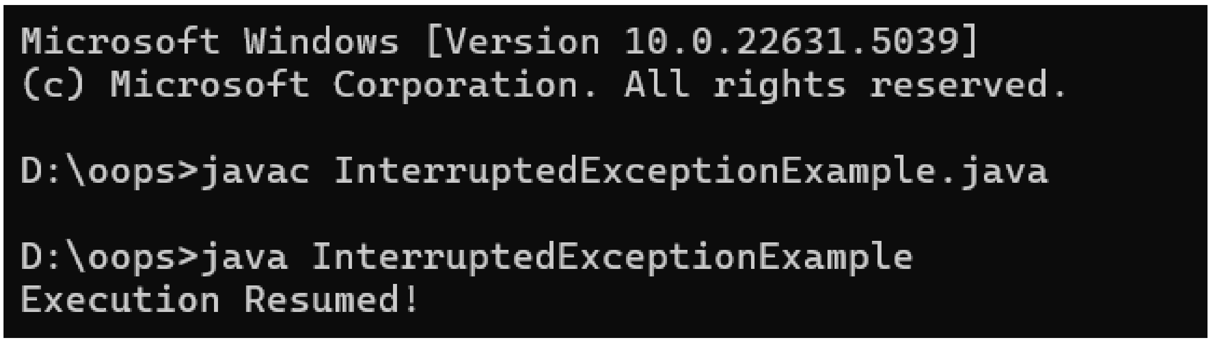
16b) InterruptedExceptionExample

Code:



```java
public class InterruptedExceptionExample {
    public static void main(String[] args) {
        try {
            Thread.sleep(1000);
            System.out.println("Execution Resumed!");
        } catch (InterruptedException e) {
            System.out.println("Thread Interrupted!");
        }
    }
}
```

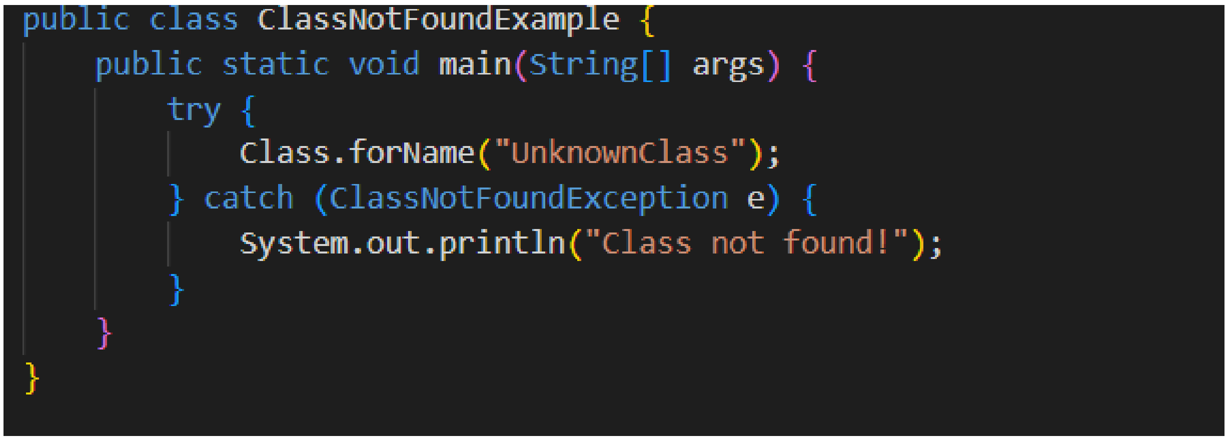
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac InterruptedExceptionExample.java

D:\oops>java InterruptedExceptionExample
Execution Resumed!
```

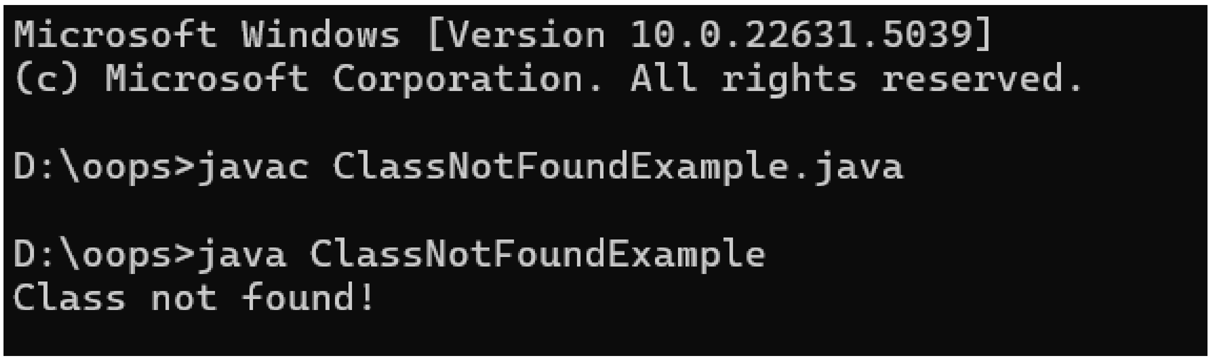
16c) ClassNotFoundExample

Code:



```java
public class ClassNotFoundExample {
    public static void main(String[] args) {
        try {
            Class.forName("UnknownClass");
        } catch (ClassNotFoundException e) {
            System.out.println("Class not found!");
        }
    }
}
```

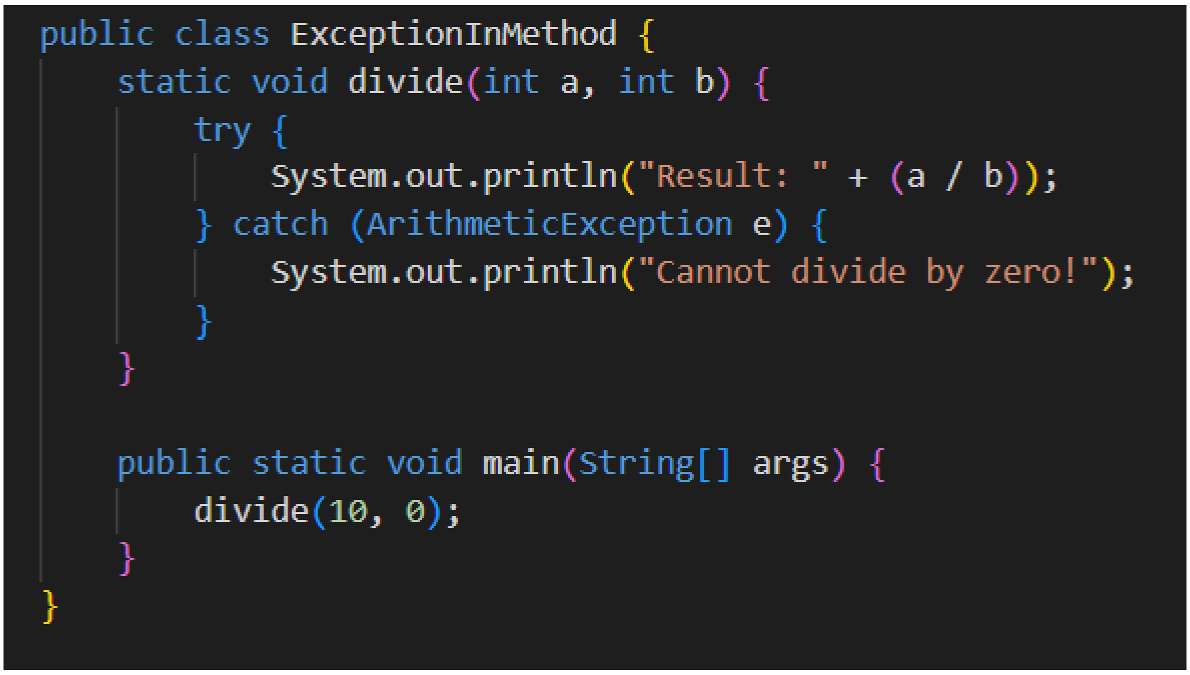
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac ClassNotFoundExample.java

D:\oops>java ClassNotFoundExample
Class not found!
```

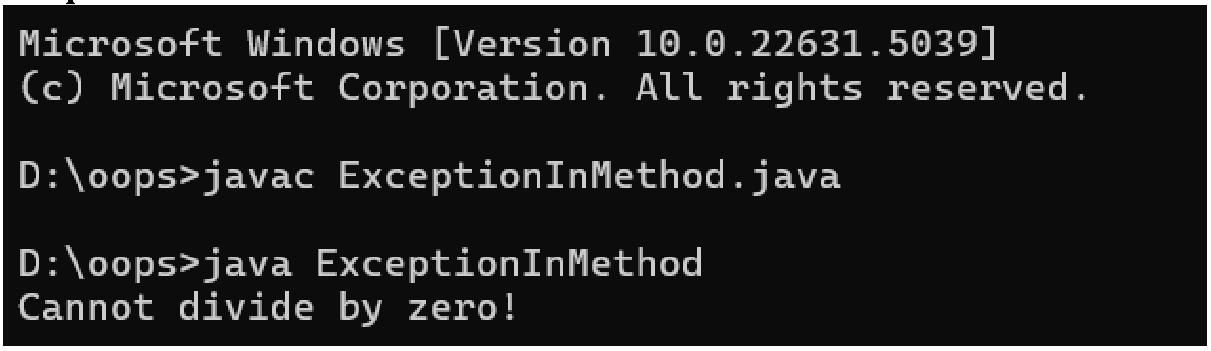
16d) ExceptionInMethod

Code:



```java
public class ExceptionInMethod {
    static void divide(int a, int b) {
        try {
            System.out.println("Result: " + (a / b));
        } catch (ArithmeticException e) {
            System.out.println("Cannot divide by zero!");
        }
    }

    public static void main(String[] args) {
        divide(10, 0);
    }
}
```

Output:

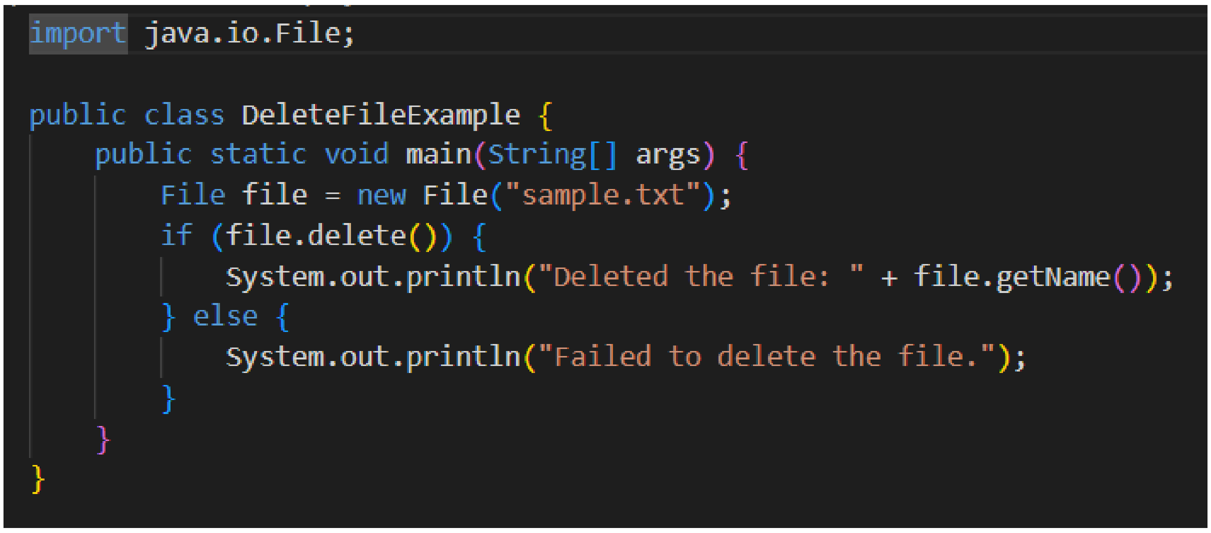


```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac ExceptionInMethod.java

D:\oops>java ExceptionInMethod
Cannot divide by zero!
```

17)FILE HANDLING PROGRAMS

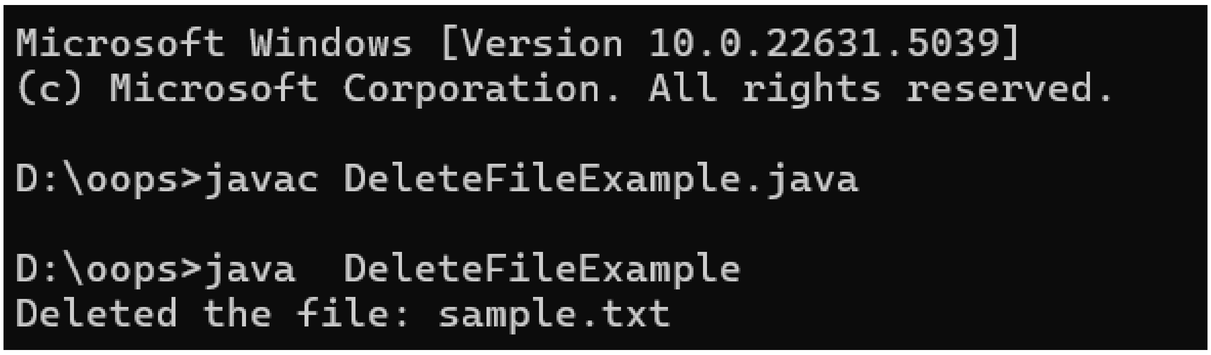
17a) DeleteFileExample

Code:



```
import java.io.File;

public class DeleteFileExample {
    public static void main(String[] args) {
        File file = new File("sample.txt");
        if (file.delete()) {
            System.out.println("Deleted the file: " + file.getName());
        } else {
            System.out.println("Failed to delete the file.");
        }
    }
}
```

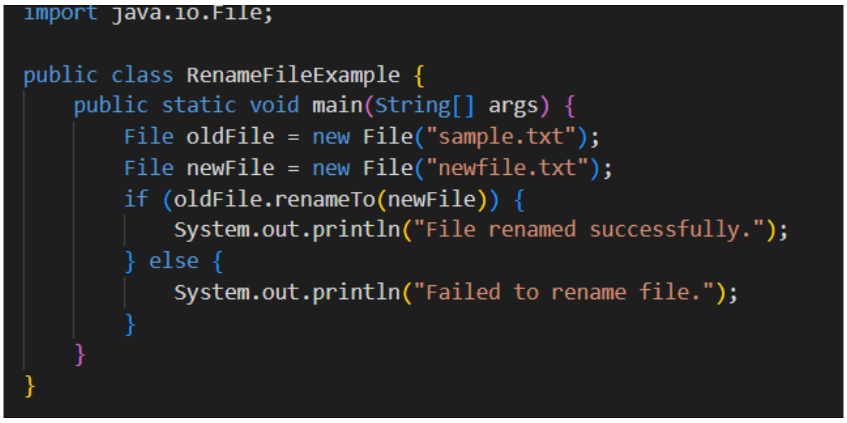
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac DeleteFileExample.java

D:\oops>java  DeleteFileExample
Deleted the file: sample.txt
```

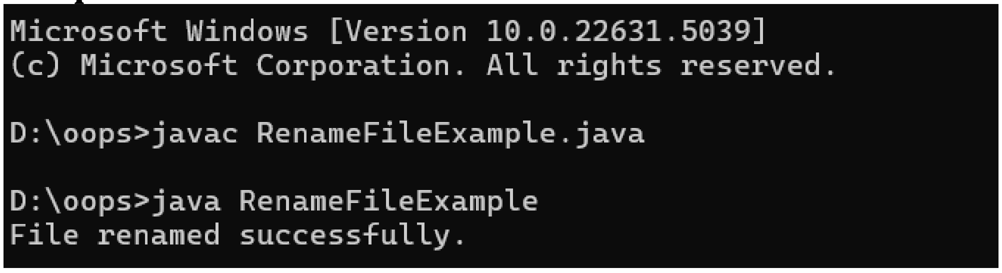
17b) RenameFileExample

Code:



```java
import java.io.File;

public class RenameFileExample {
    public static void main(String[] args) {
        File oldFile = new File("sample.txt");
        File newFile = new File("newfile.txt");
        if (oldFile.renameTo(newFile)) {
            System.out.println("File renamed successfully.");
        } else {
            System.out.println("Failed to rename file.");
        }
    }
}
```

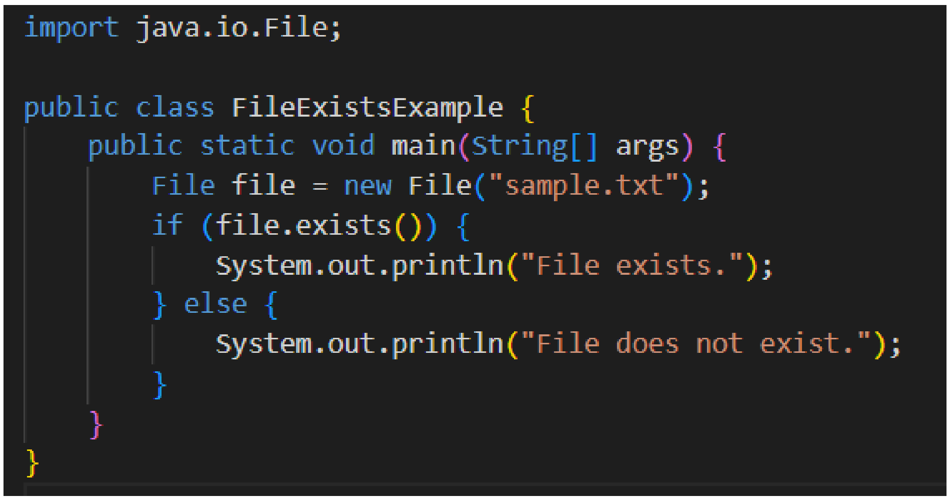
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac RenameFileExample.java

D:\oops>java RenameFileExample
File renamed successfully.
```

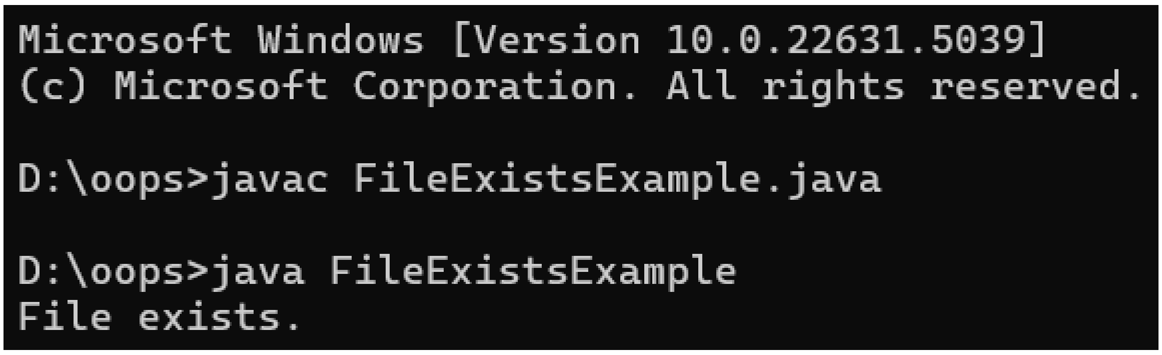
17c) FileExistExample

Code:



```
import java.io.File;

public class FileExistsExample {
    public static void main(String[] args) {
        File file = new File("sample.txt");
        if (file.exists()) {
            System.out.println("File exists.");
        } else {
            System.out.println("File does not exist.");
        }
    }
}
```

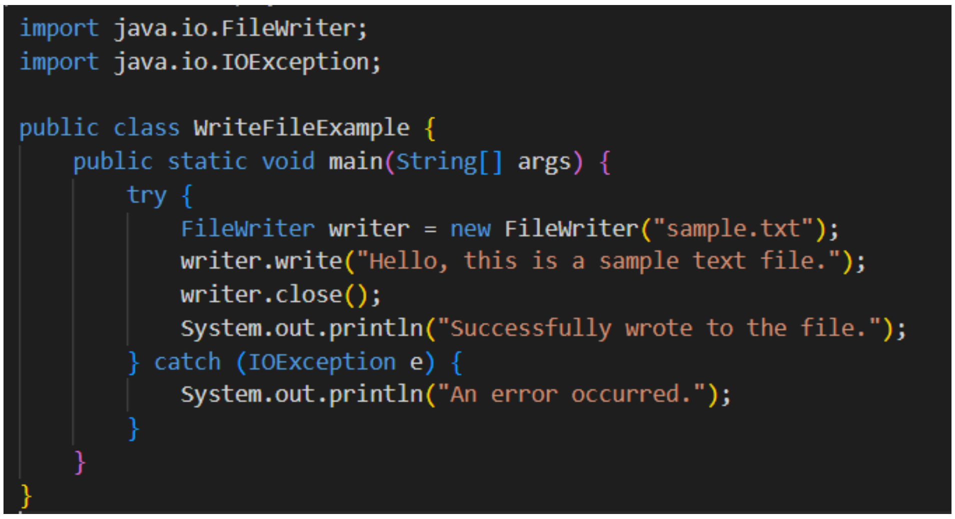
Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac FileExistsExample.java

D:\oops>java FileExistsExample
File exists.
```

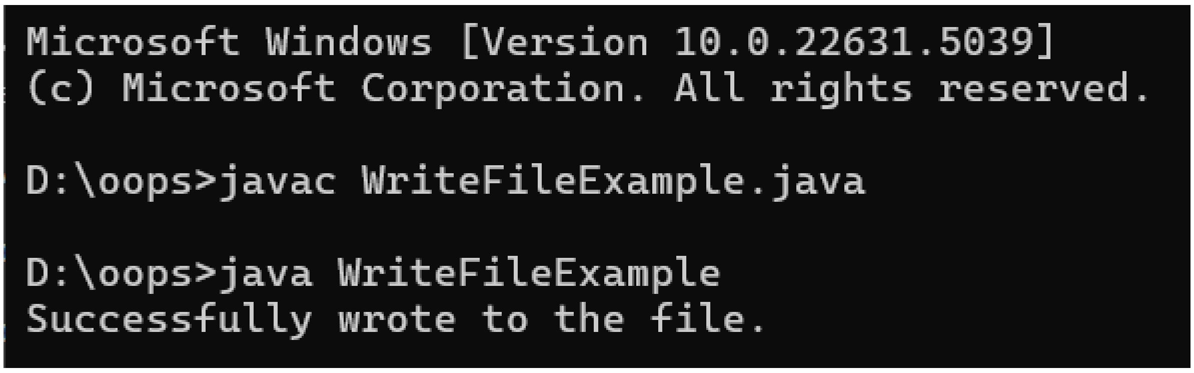
17d) WriteFileExample

Code:



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

public class WriteFileExample {
    public static void main(String[] args) {
        try {
            FileWriter writer = new FileWriter("sample.txt");
            writer.write("Hello, this is a sample text file.");
            writer.close();
            System.out.println("Successfully wrote to the file.");
        } catch (IOException e) {
            System.out.println("An error occurred.");
        }
    }
}
```

Output:



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

D:\oops>javac WriteFileExample.java

D:\oops>java WriteFileExample
Successfully wrote to the file.
```