LAB RECORD

23CSE111- Object Oriented Programming

Submitted by

CH.SC.U4CSE24132 - N.Preethi Jasmine

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.U4CSE24132 – N.Preethi Jasmine*** 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

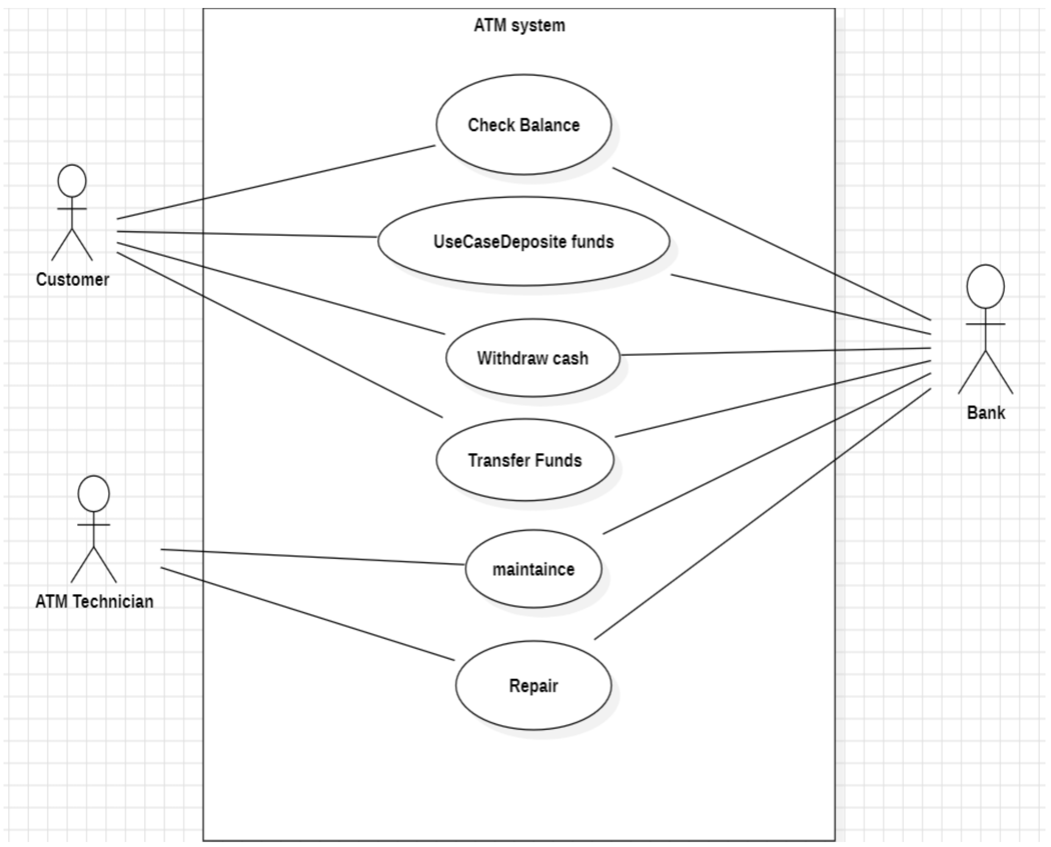
Internal Examiner 1 Internal Examiner 2

INDEX

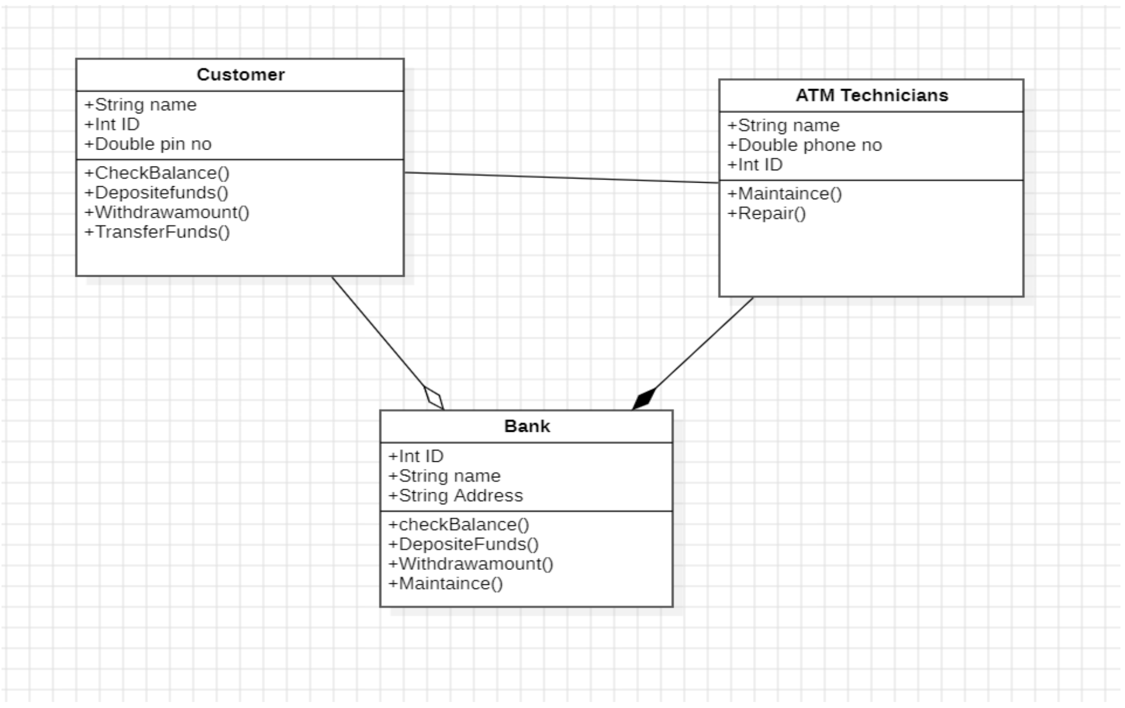
<u>UML DIAGRAMS</u>

1.ATM SYSTEM

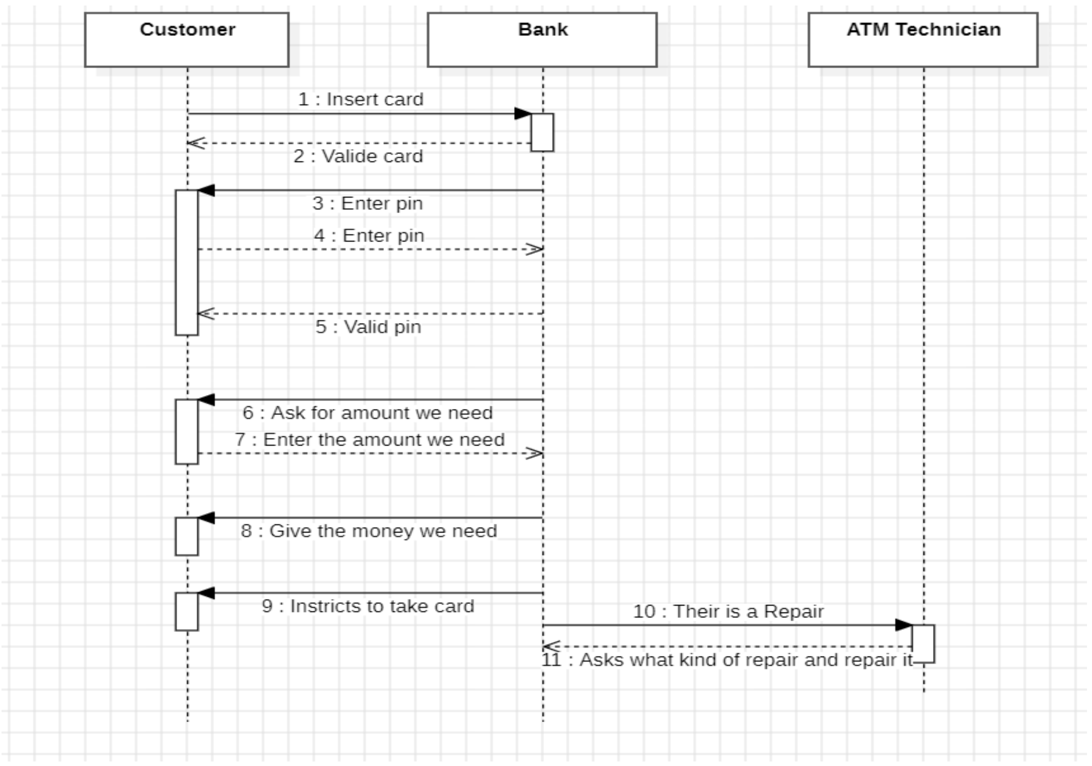
1(a): USE CASE DIAGRAM



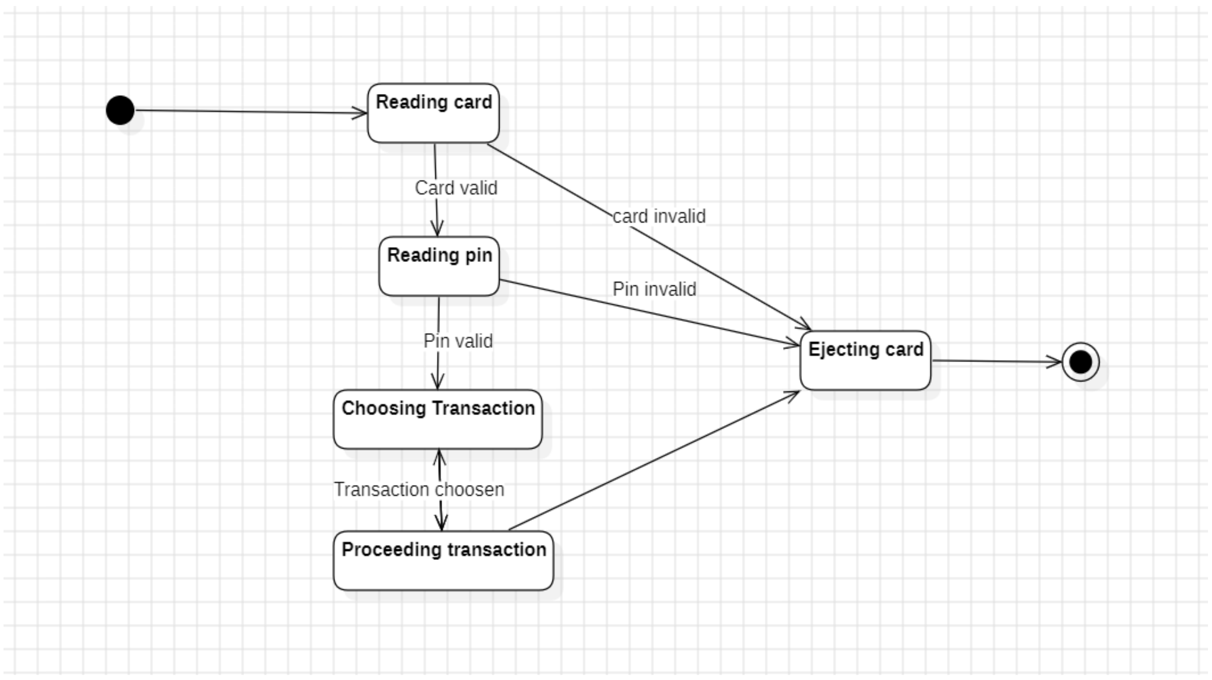
1(b): CLASS DIAGRAM



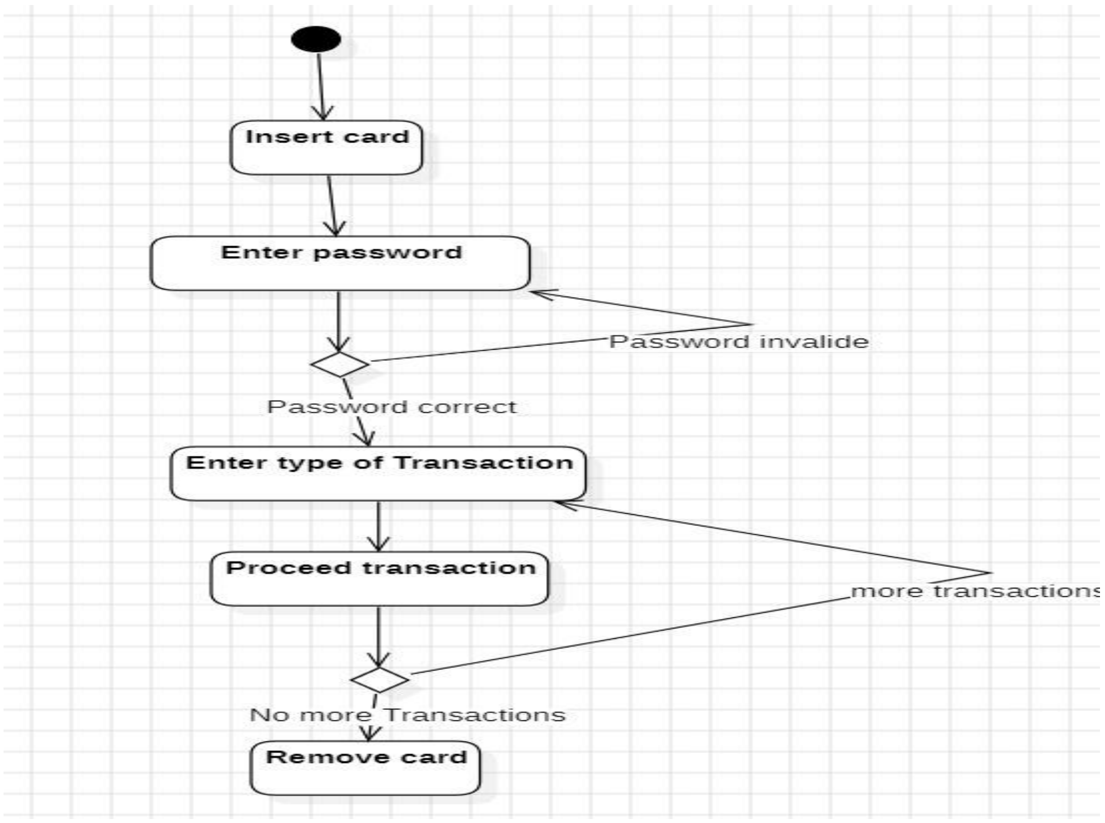
1(c): SEQUENCE DIAGRAM


Customer
Bank
ATM Technician
1 : Insert card
2 : Valide card
3 : Enter pin
4 : Enter pin
5 : Valid pin
6 : Ask for amount we need
7 : Enter the amount we need
8 : Give the money we need
9 : Instricts to take card
10 : Their is a Repair
11 : Asks what kind of repair and repair it

1(d): STATE DIAGRAM

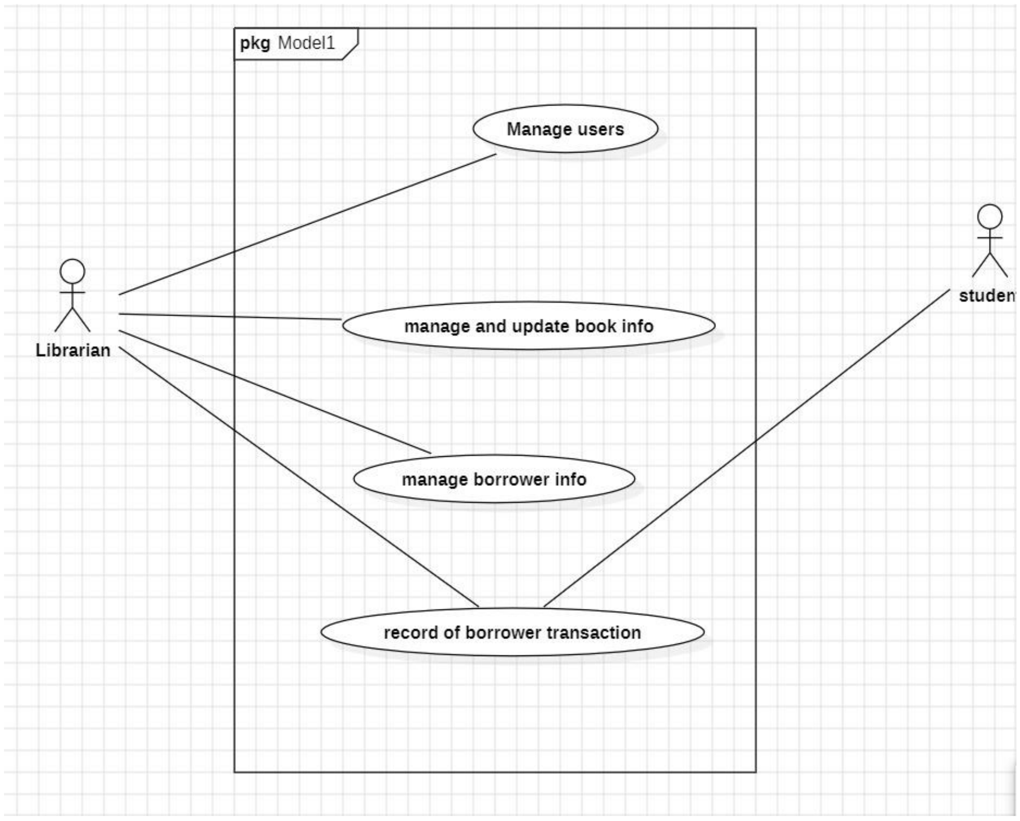


1(e): ACTIVITY DIAGRAM

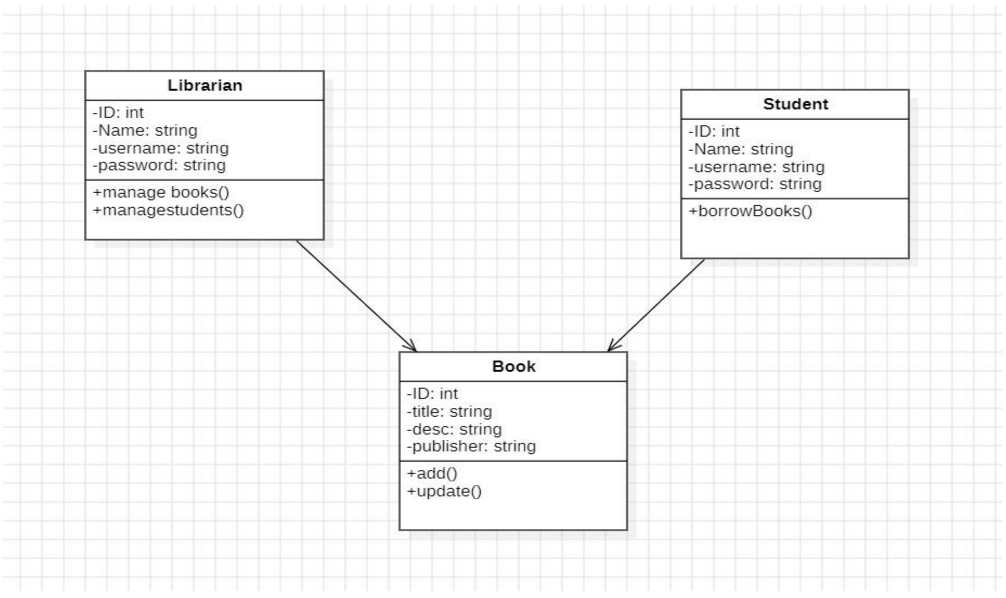


2.ONLINE ATTENDENCE

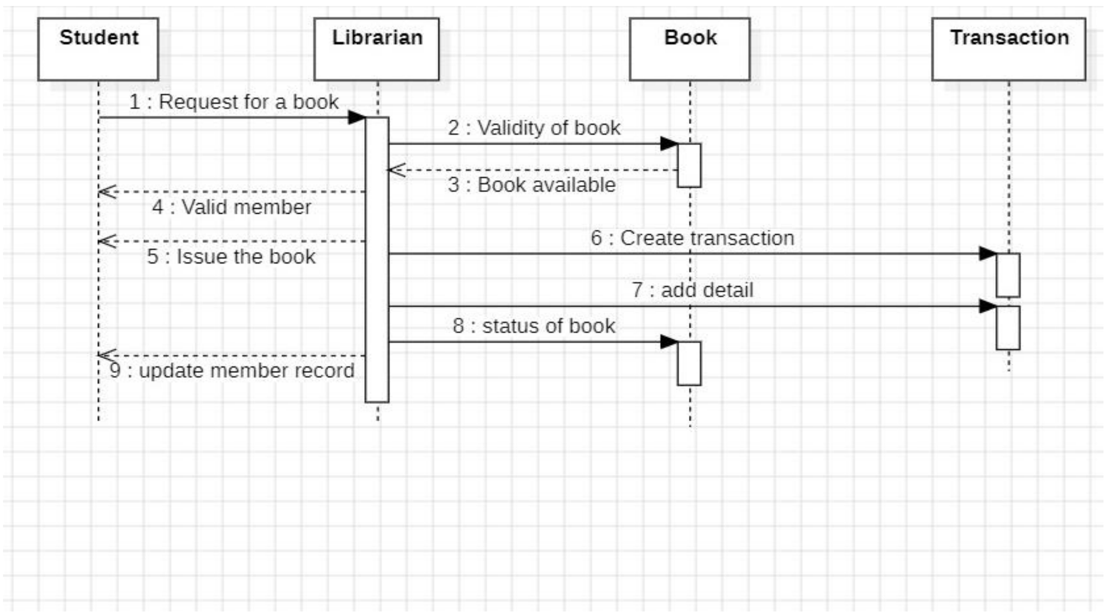
2(A): USE CASE DIAGRAM



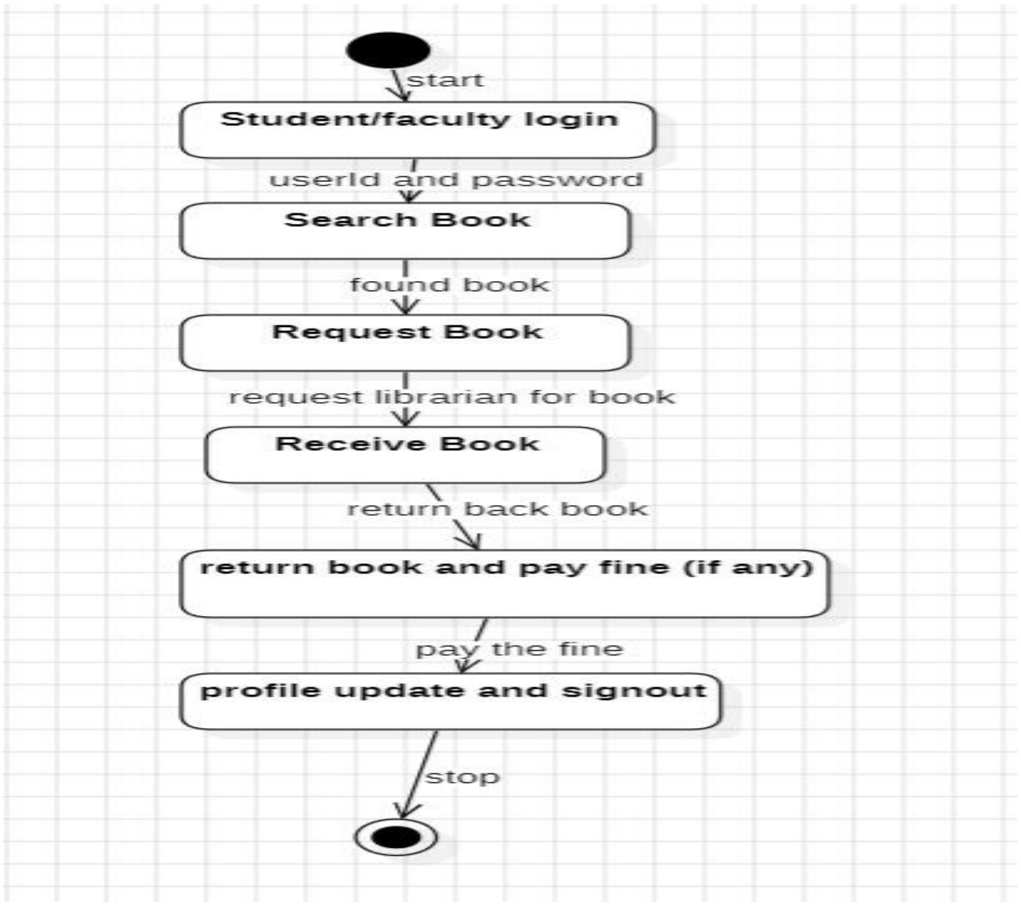
2(B): CLASS DIAGRAM



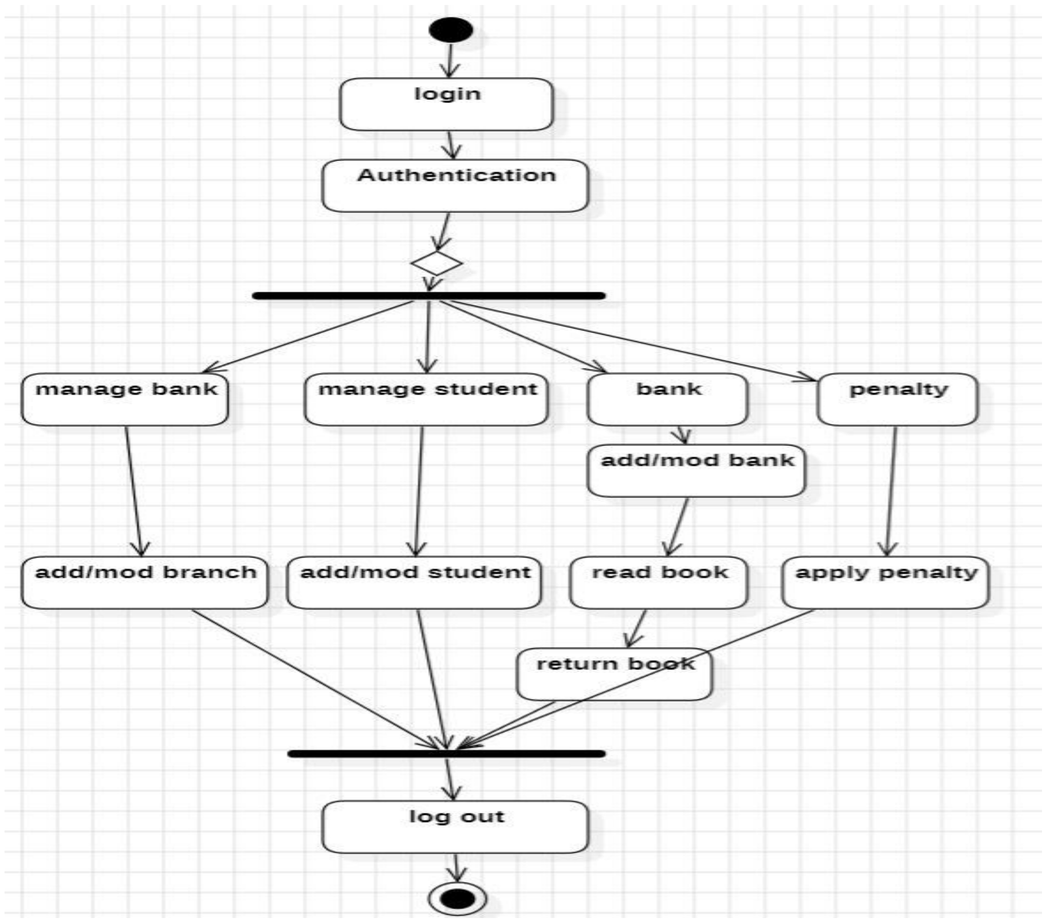
2(C): SEQUENCE DIAGRAM



2(D): STATE DIAGRAM



2(E): ACTIVITY DIAGRAM



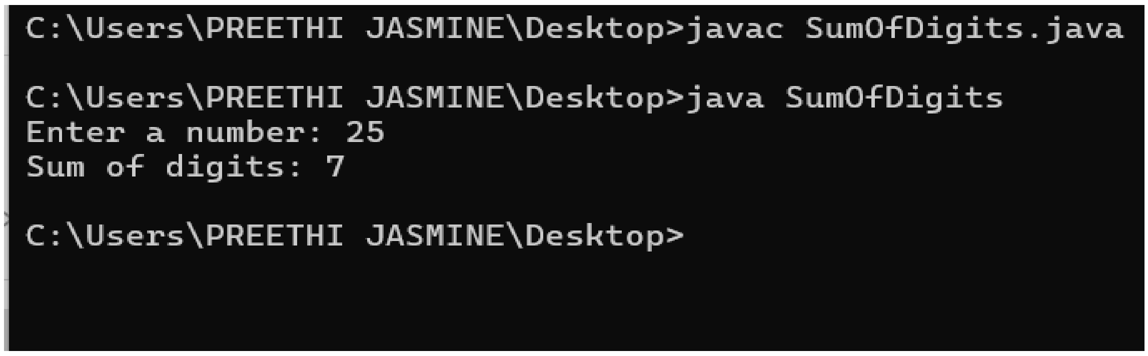
3. JAVA BASIC PROGRAMS

3(a): SUM OF DIGITS

CODE:

```
import java.util.Scanner;

public class SumOfDigits {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        int sum = 0;

        while (num != 0) {
            sum += num % 10;
            num /= 10;
        }

        System.out.println("Sum of digits: " + sum);
        sc.close();
    }
}
```

OUTPUT:



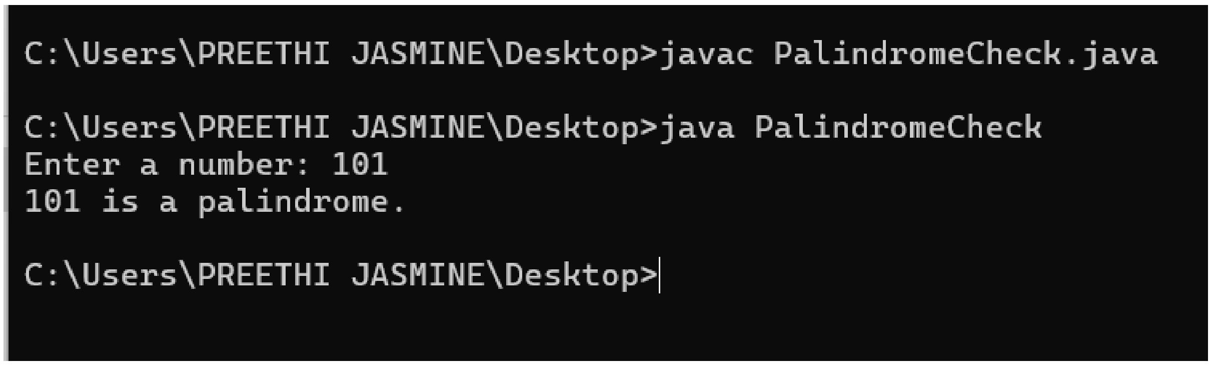
```
C:\Users\PREETHI JASMINE\Desktop>javac SumOfDigits.java

C:\Users\PREETHI JASMINE\Desktop>java SumOfDigits
Enter a number: 25
Sum of digits: 7

C:\Users\PREETHI JASMINE\Desktop>
```

3(b):PalindromeCheck

CODE:

```
import java.util.Scanner;

public class PalindromeCheck {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        int original = num, reversed = 0;

        while (num != 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }

        if (original == reversed)
            System.out.println(original + " is a palindrome.");
        else
            System.out.println(original + " is not a palindrome.");
        sc.close();
    }
}
```

OUTPUT:



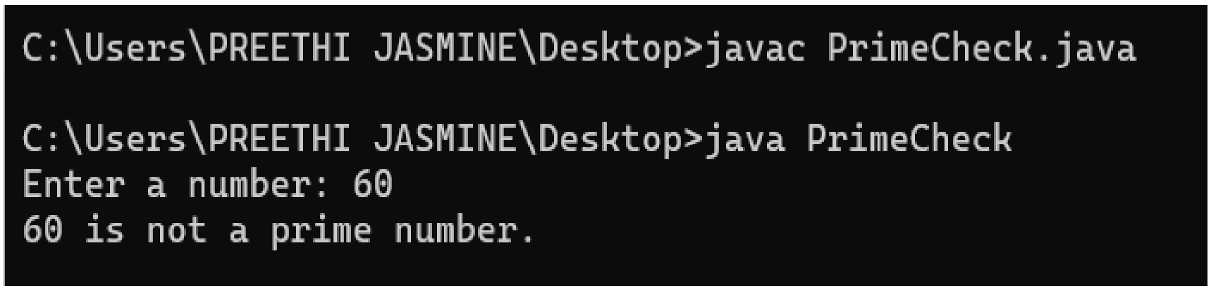
```
C:\Users\PREETHI JASMINE\Desktop>javac PalindromeCheck.java

C:\Users\PREETHI JASMINE\Desktop>java PalindromeCheck
Enter a number: 101
101 is a palindrome.

C:\Users\PREETHI JASMINE\Desktop>
```

3(c): Check Prime Number

CODE:

```
import java.util.Scanner;

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

        if (num <= 1)
            isPrime = false;
        else {
            for (int i = 2; i <= Math.sqrt(num); i++) {
                if (num % i == 0) {
                    is Prime = false;
                    break;
                }
            }
        }

        if (isPrime)
            System.out.println(num + " is a prime number.");
        else
            System.out.println(num + " is not a prime number.");

        sc.close();
    }
}
```

OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac PrimeCheck.java

C:\Users\PREETHI JASMINE\Desktop>java PrimeCheck
Enter a number: 60
60 is not a prime number.
```

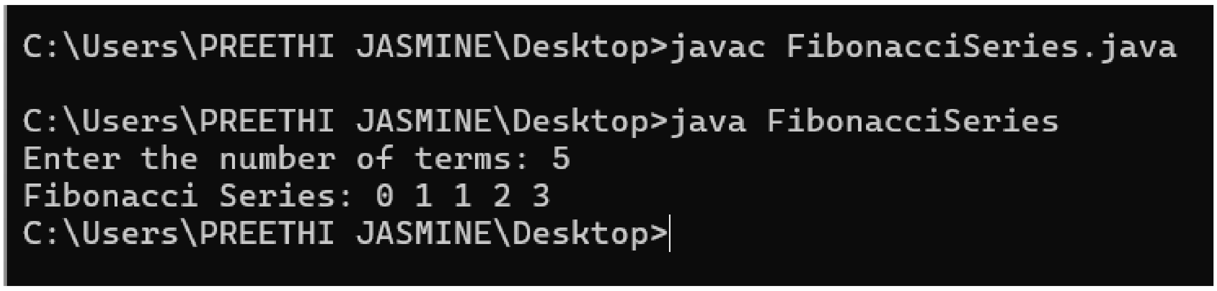
3(d): Fibonacci Numbers

CODE:

```
import java.util.Scanner;

public class FibonacciSeries {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter the number of terms: ");
    int n = sc.nextInt();
```

```
        int a = 0, b = 1, next;
        System.out.print("Fibonacci Series: " + a + " " + b);

        for (int i = 2; i < n; i++) {
            next = a + b;
            System.out.print(" " + next);
            a = b;
            b = next;
        }

        sc.close();
    }
}
```

OUTPUT:



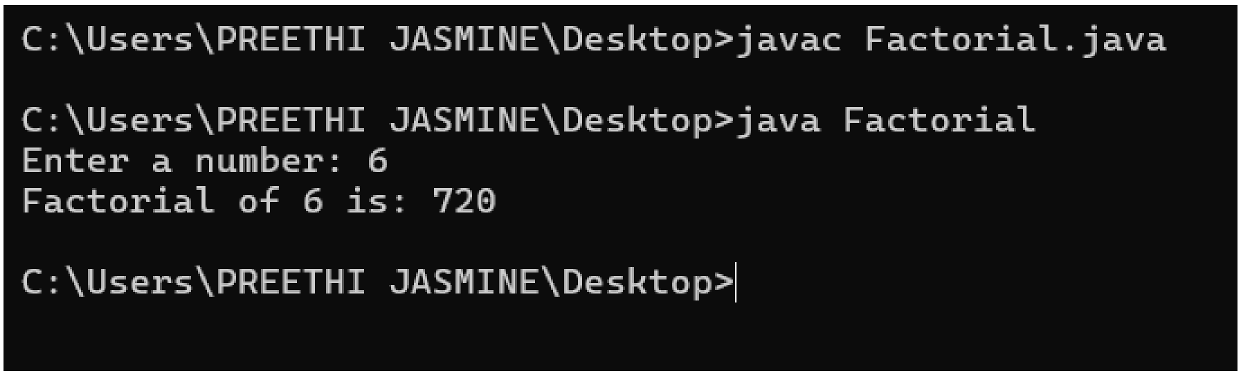
```
C:\Users\PREETHI JASMINE\Desktop>javac FibonacciSeries.java

C:\Users\PREETHI JASMINE\Desktop>java FibonacciSeries
Enter the number of terms: 5
Fibonacci Series: 0 1 1 2 3
C:\Users\PREETHI JASMINE\Desktop>
```

3(e): Factorial Of a Number

CODE:

```
import java.util.Scanner;

public class Factorial {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        int fact = 1;

        for (int i = 1; i <= num; i++) {
            fact *= i;
        }

        System.out.println("Factorial of " + num + " is: " + fact);
        sc.close();
    }
}
```

OUTPUT:



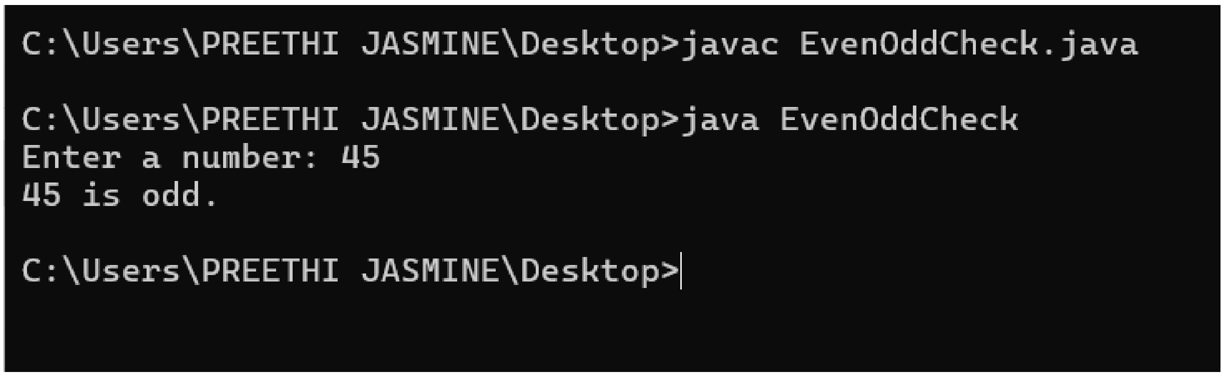
```
C:\Users\PREETHI JASMINE\Desktop>javac Factorial.java

C:\Users\PREETHI JASMINE\Desktop>java Factorial
Enter a number: 6
Factorial of 6 is: 720

C:\Users\PREETHI JASMINE\Desktop>
```

3(f): Check Even Or Odd

CODE:

```
import java.util.Scanner;

public class EvenOddCheck {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        if (num % 2 == 0)
            System.out.println(num + " is even.");
        else
            System.out.println(num + " is odd.");

        sc.close();
    }
}
```

OUTPUT:



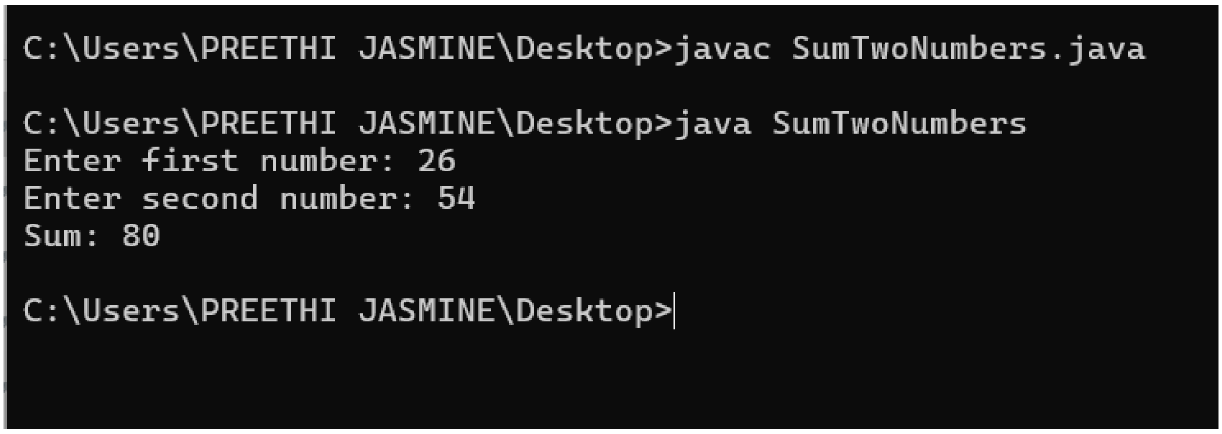
```
C:\Users\PREETHI JASMINE\Desktop>javac EvenOddCheck.java

C:\Users\PREETHI JASMINE\Desktop>java EvenOddCheck
Enter a number: 45
45 is odd.

C:\Users\PREETHI JASMINE\Desktop>
```

3(g): Sum Of Two Numbers

CODE:

```
import java.util.Scanner;

public class SumTwoNumbers {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter first number: ");
        int num1 = sc.nextInt();
        System.out.print("Enter second number: ");
        int num2 = sc.nextInt();
        int sum = num1 + num2;
        System.out.println("Sum: " + sum);
        sc.close();
    }
}
```

OUTPUT:



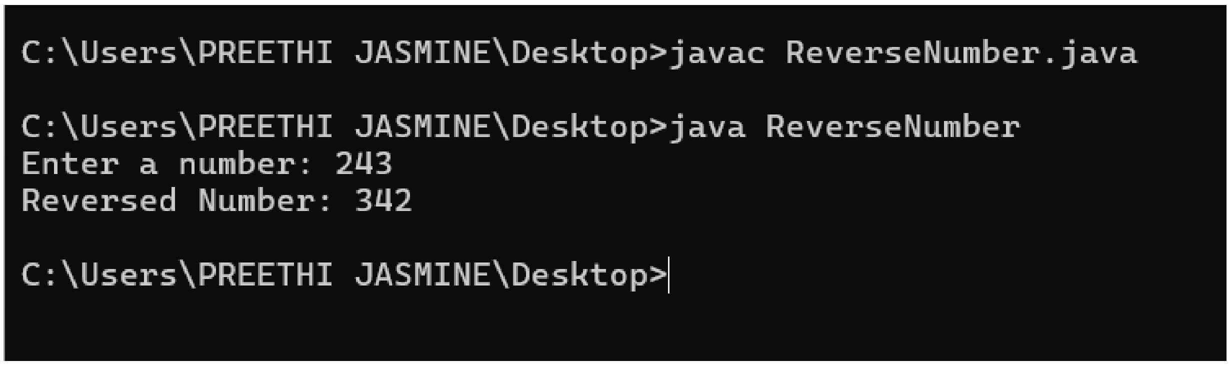
```
C:\Users\PREETHI JASMINE\Desktop>javac SumTwoNumbers.java

C:\Users\PREETHI JASMINE\Desktop>java SumTwoNumbers
Enter first number: 26
Enter second number: 54
Sum: 80

C:\Users\PREETHI JASMINE\Desktop>
```

3(h): Reverse a Number

CODE:

```
import java.util.Scanner;

public class ReverseNumber {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        int reversed = 0;

        while (num != 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }

        System.out.println("Reversed Number: " + reversed);
        sc.close();
    }
}
```

OUTPUT:



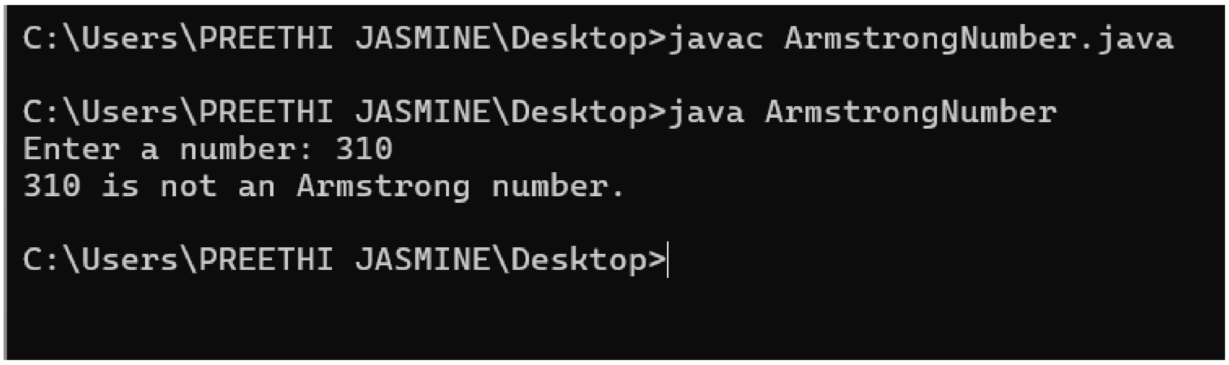
```
C:\Users\PREETHI JASMINE\Desktop>javac ReverseNumber.java

C:\Users\PREETHI JASMINE\Desktop>java ReverseNumber
Enter a number: 243
Reversed Number: 342

C:\Users\PREETHI JASMINE\Desktop>
```

3(i): Armstrong Number

CODE:

```
import java.util.Scanner;

public class ArmstrongNumber {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        int original = num, sum = 0, digits = 0, temp = num;

        while (temp != 0) {
            temp /= 10;
            digits++;
        }

        temp = num;
        while (temp != 0) {
            int digit = temp % 10;
            sum += Math.pow(digit, digits);
            temp /= 10;
        }

        if (sum == original)
            System.out.println(original + " is an Armstrong number.");
        else
            System.out.println(original + " is not an Armstrong number.");

        sc.close();
    }
}
```

OUTPUT:



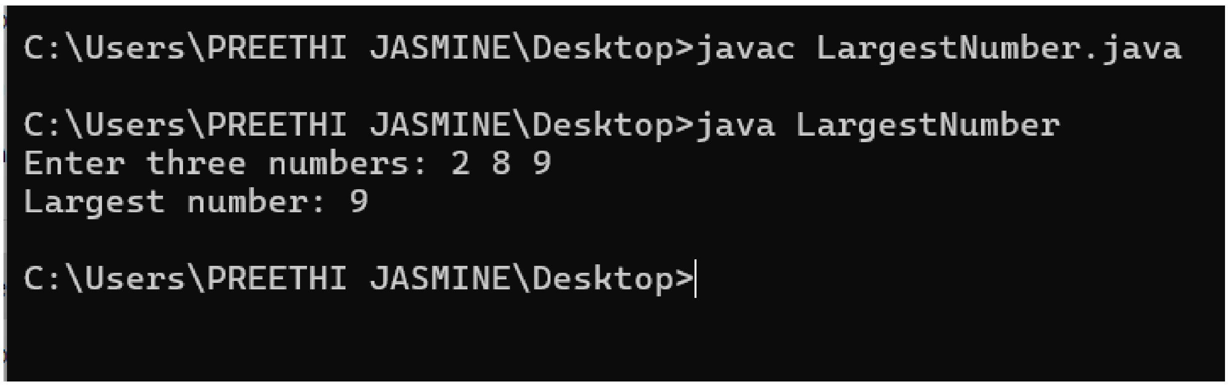
```
C:\Users\PREETHI JASMINE\Desktop>javac ArmstrongNumber.java

C:\Users\PREETHI JASMINE\Desktop>java ArmstrongNumber
Enter a number: 310
310 is not an Armstrong number.

C:\Users\PREETHI JASMINE\Desktop>
```

3(j): Find The Largest Number

CODE:

```java
import java.util.Scanner;

public class LargestNumber {
    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();

        int largest = (a > b) ? (a > c ? a : c) : (b > c ? b : c);
        System.out.println("Largest number: " + largest);

        sc.close();
    }
}
```

OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac LargestNumber.java

C:\Users\PREETHI JASMINE\Desktop>java LargestNumber
Enter three numbers: 2 8 9
Largest number: 9

C:\Users\PREETHI JASMINE\Desktop>
```

INHERITENCE

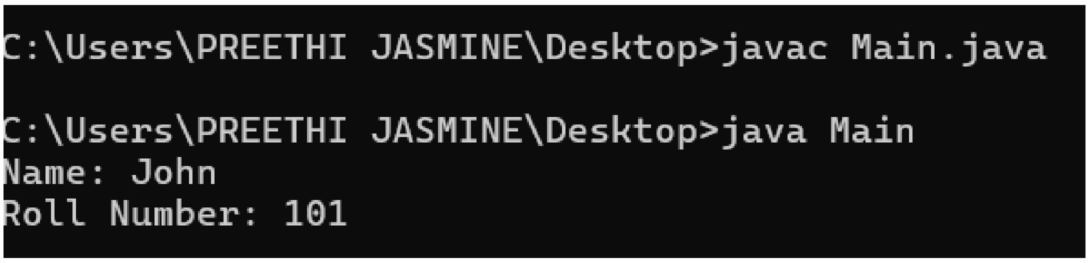
4)SINGLE INHERITANCE PROGRAMS

A) Student details

CODE:

```
class Person {
    String name = "John";

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

class Student extends Person {
    int rollNumber = 101;

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

public class Main {
    public static void main(String[] args) {
        Student s = new Student();
        s.displayName();
        s.displayRollNumber();
    }
}
```

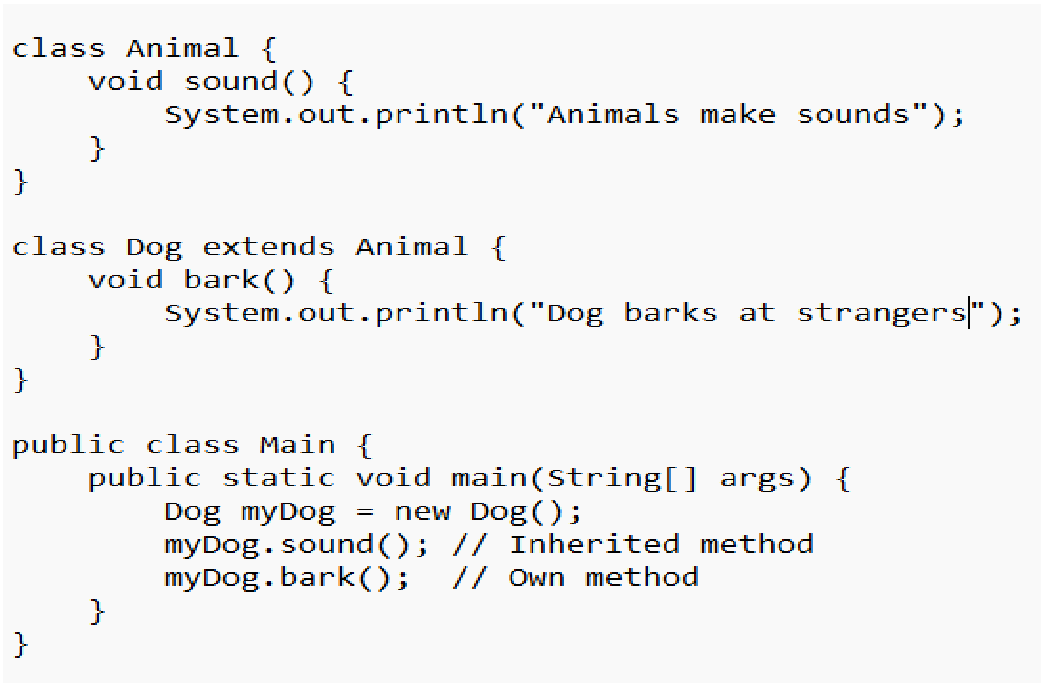
OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac Main.java

C:\Users\PREETHI JASMINE\Desktop>java Main
Name: John
Roll Number: 101
```

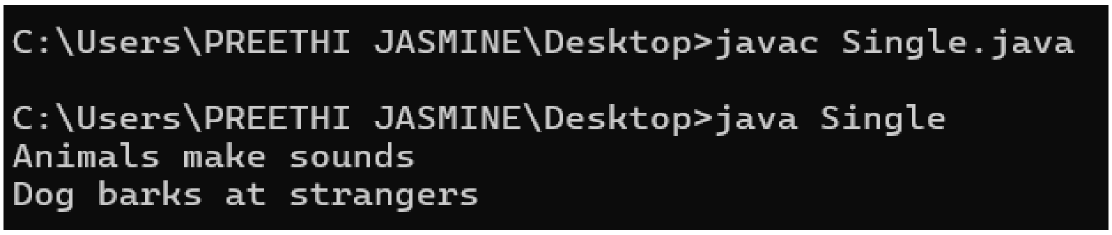
B)Animal Sounds

CODE:



```java
class Animal {
    void sound() {
        System.out.println("Animals make sounds");
    }
}

class Dog extends Animal {
    void bark() {
        System.out.println("Dog barks at strangers");
    }
}

public class Main {
    public static void main(String[] args) {
        Dog myDog = new Dog();
        myDog.sound(); // Inherited method
        myDog.bark();  // Own method
    }
}
```

OUTPUT:

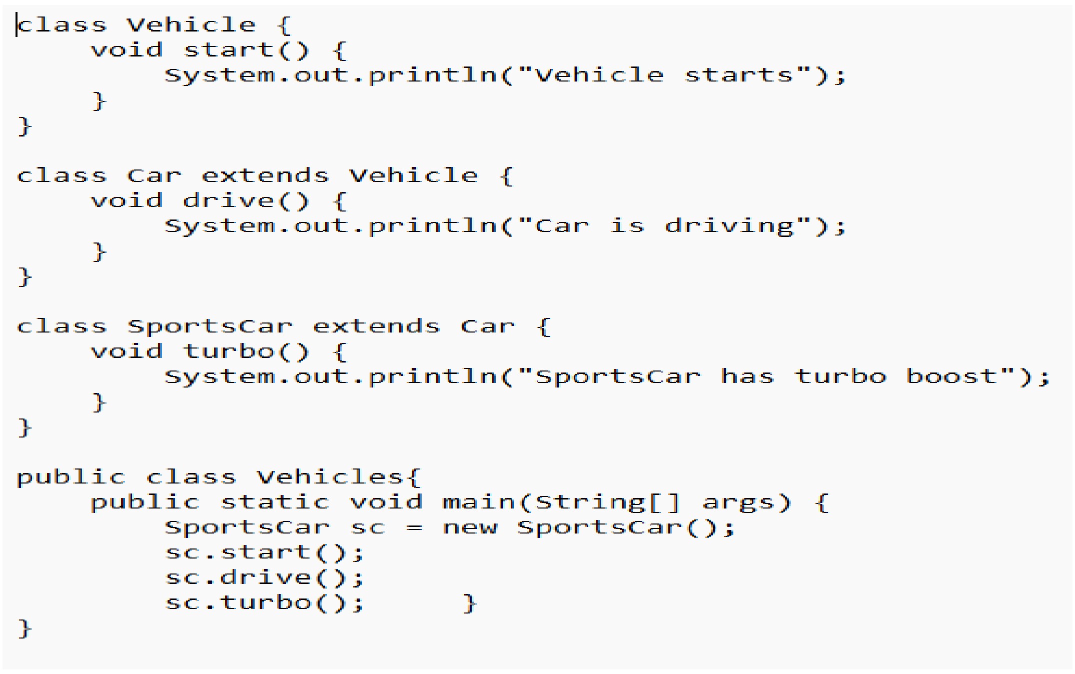


```
C:\Users\PREETHI JASMINE\Desktop>javac Single.java

C:\Users\PREETHI JASMINE\Desktop>java Single
Animals make sounds
Dog barks at strangers
```

5)MULTILEVEL INHERITENCE PROGRAMS

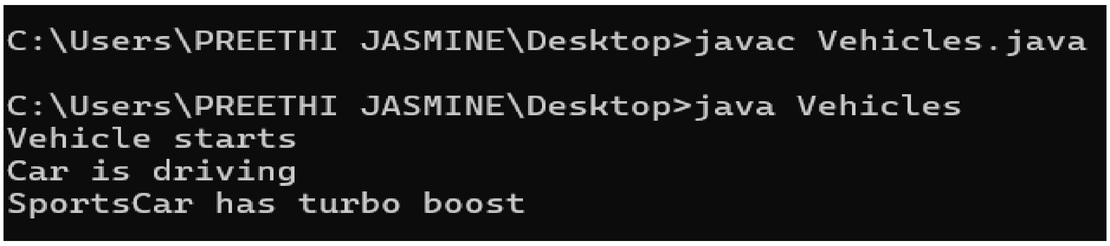
A) Vehicles

CODE:



```
class Vehicle {
    void start() {
        System.out.println("Vehicle starts");
    }
}

class Car extends Vehicle {
    void drive() {
        System.out.println("Car is driving");
    }
}

class SportsCar extends Car {
    void turbo() {
        System.out.println("SportsCar has turbo boost");
    }
}

public class Vehicles{
    public static void main(String[] args) {
        SportsCar sc = new SportsCar();
        sc.start();
        sc.drive();
        sc.turbo();     }
}
```

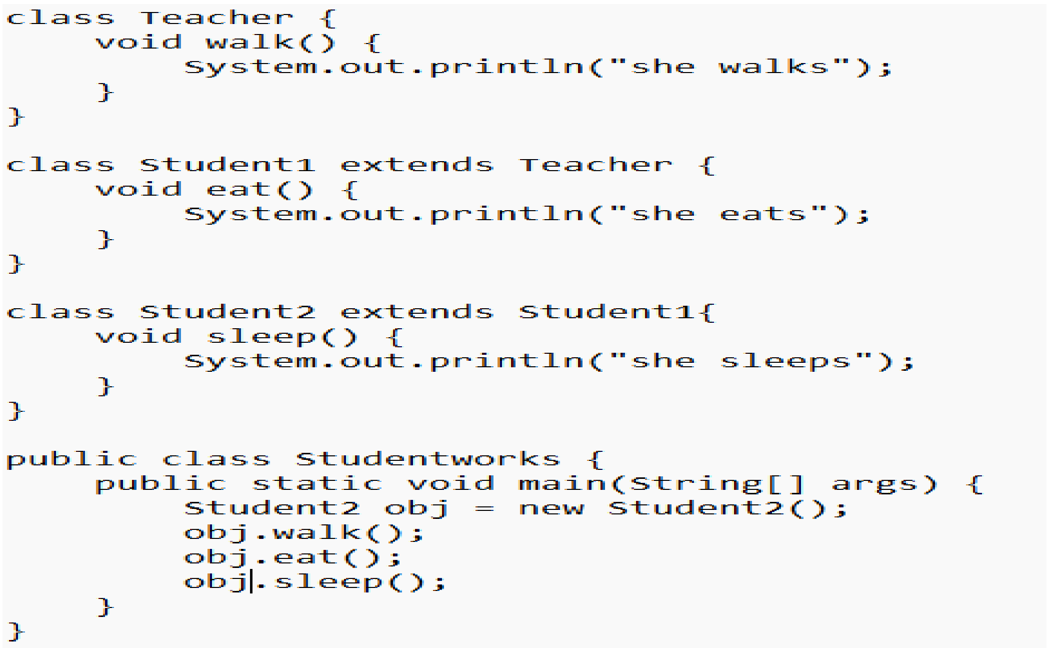
OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac Vehicles.java

C:\Users\PREETHI JASMINE\Desktop>java Vehicles
Vehicle starts
Car is driving
SportsCar has turbo boost
```

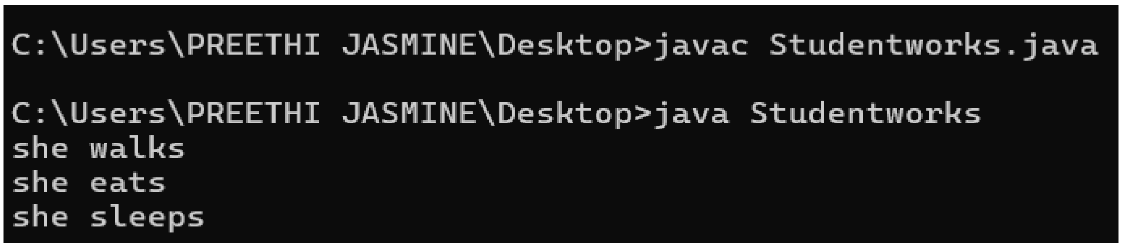
B) Studentworks

CODE:



```
class Teacher {
    void walk() {
        System.out.println("she walks");
    }
}

class Student1 extends Teacher {
    void eat() {
        System.out.println("she eats");
    }
}

class Student2 extends Student1{
    void sleep() {
        System.out.println("she sleeps");
    }
}

public class Studentworks {
    public static void main(String[] args) {
        Student2 obj = new Student2();
        obj.walk();
        obj.eat();
        obj.sleep();
    }
}
```

OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac Studentworks.java

C:\Users\PREETHI JASMINE\Desktop>java Studentworks
she walks
she eats
she sleeps
```

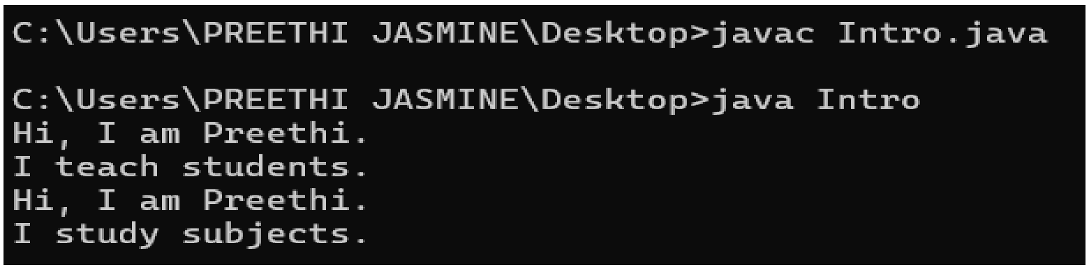
6) HIERARCHICAL INHERITANCE PROGRAMS

A) Introduction

CODE:

```java
class Person {
    void introduce() {
        System.out.println("Hi, I am Preethi.");
    }
}

class Teacher extends Person {
    void teach() {
        System.out.println("I teach students.");
    }
}

class Student extends Person {
    void study() {
        System.out.println("I study subjects.");
    }
}

public class Intro{
    public static void main(String[] args) {
        Teacher t = new Teacher();
        t.introduce();
        t.teach();

        Student s = new Student();
        s.introduce();
        s.study();
    }
}
```

OUTPUT:



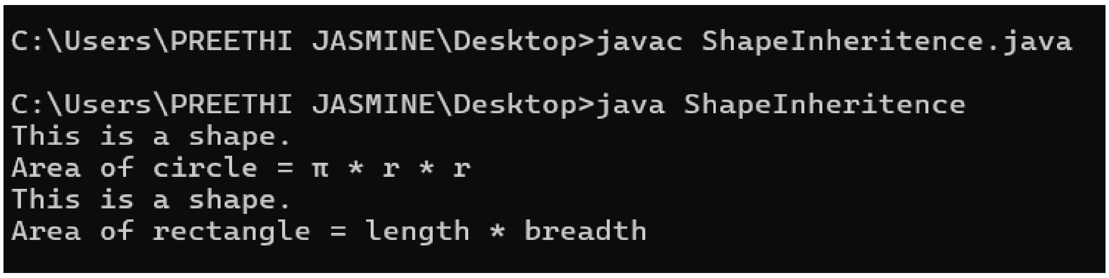
```
C:\Users\PREETHI JASMINE\Desktop>javac Intro.java

C:\Users\PREETHI JASMINE\Desktop>java Intro
Hi, I am Preethi.
I teach students.
Hi, I am Preethi.
I study subjects.
```

B) ShapeInheritence

CODE:

```
class Shape {
    void display() {
        System.out.println("This is a shape.");
    }
}

class Circle extends Shape {
    void area() {
        System.out.println("Area of circle = π * r * r");
    }
}

class Rectangle extends Shape {
    void area() {
        System.out.println("Area of rectangle = length * breadth");
    }
}

public class ShapeInheritence {
    public static void main(String[] args) {
        Circle c = new Circle();
        c.display();
        c.area();

        Rectangle r = new Rectangle();
        r.display();
        r.area();
    }
}
```

OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac ShapeInheritence.java

C:\Users\PREETHI JASMINE\Desktop>java ShapeInheritence
This is a shape.
Area of circle = π * r * r
This is a shape.
Area of rectangle = length * breadth
```

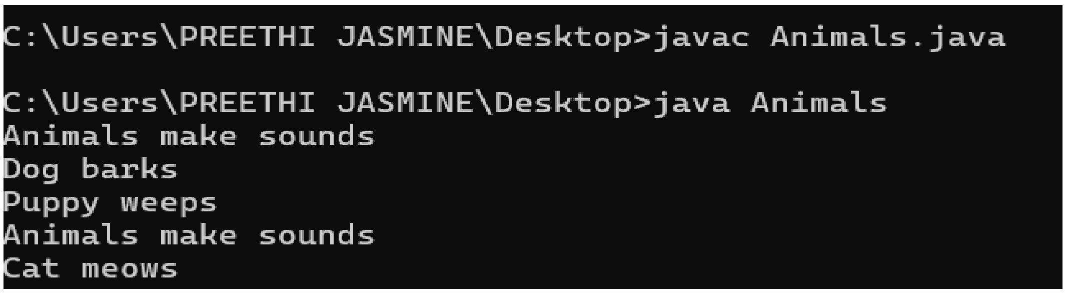
7) HYBRID INHERITENCE PROGRAMS

A) Animals

CODE:

```
class Animal {
    void sound() {
        System.out.println("Animals make sounds");
    }
}

class Dog extends Animal {
    void bark() {
        System.out.println("Dog barks");
    }
}

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

class Puppy extends Dog {
    void weep() {
        System.out.println("Puppy weeps");
    }
}

public class Animals {
    public static void main(String[] args) {
        Puppy p = new Puppy();
        p.sound();
        p.bark();
        p.weep();

        Cat c = new Cat();
        c.sound();
        c.meow();
    }
}
```

OUTPUT:



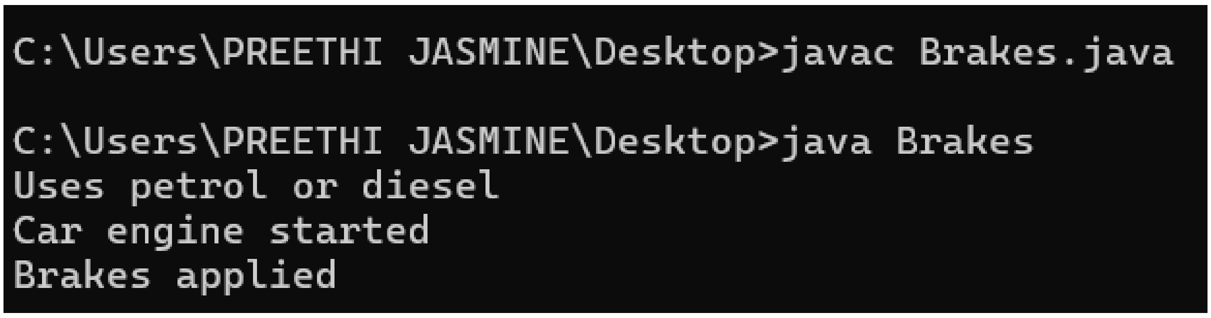
```
C:\Users\PREETHI JASMINE\Desktop>javac Animals.java

C:\Users\PREETHI JASMINE\Desktop>java Animals
Animals make sounds
Dog barks
Puppy weeps
Animals make sounds
Cat meows
```

B) Brakes

CODE:

```
interface Engine {
    void start();
}

interface Brake {
    void applyBrake();
}

class Vehicle {
    void fuelType() {
        System.out.println("Uses petrol or diesel");
    }
}

class Car extends Vehicle implements Engine, Brake {
    public void start() {
        System.out.println("Car engine started");
    }

    public void applyBrake() {
        System.out.println("Brakes applied");
    }
}

public class Brakes {
    public static void main(String[] args) {
        Car c = new Car();
        c.fuelType();
        c.start();
        c.applyBrake();
    }
}
```

OUTPUT:



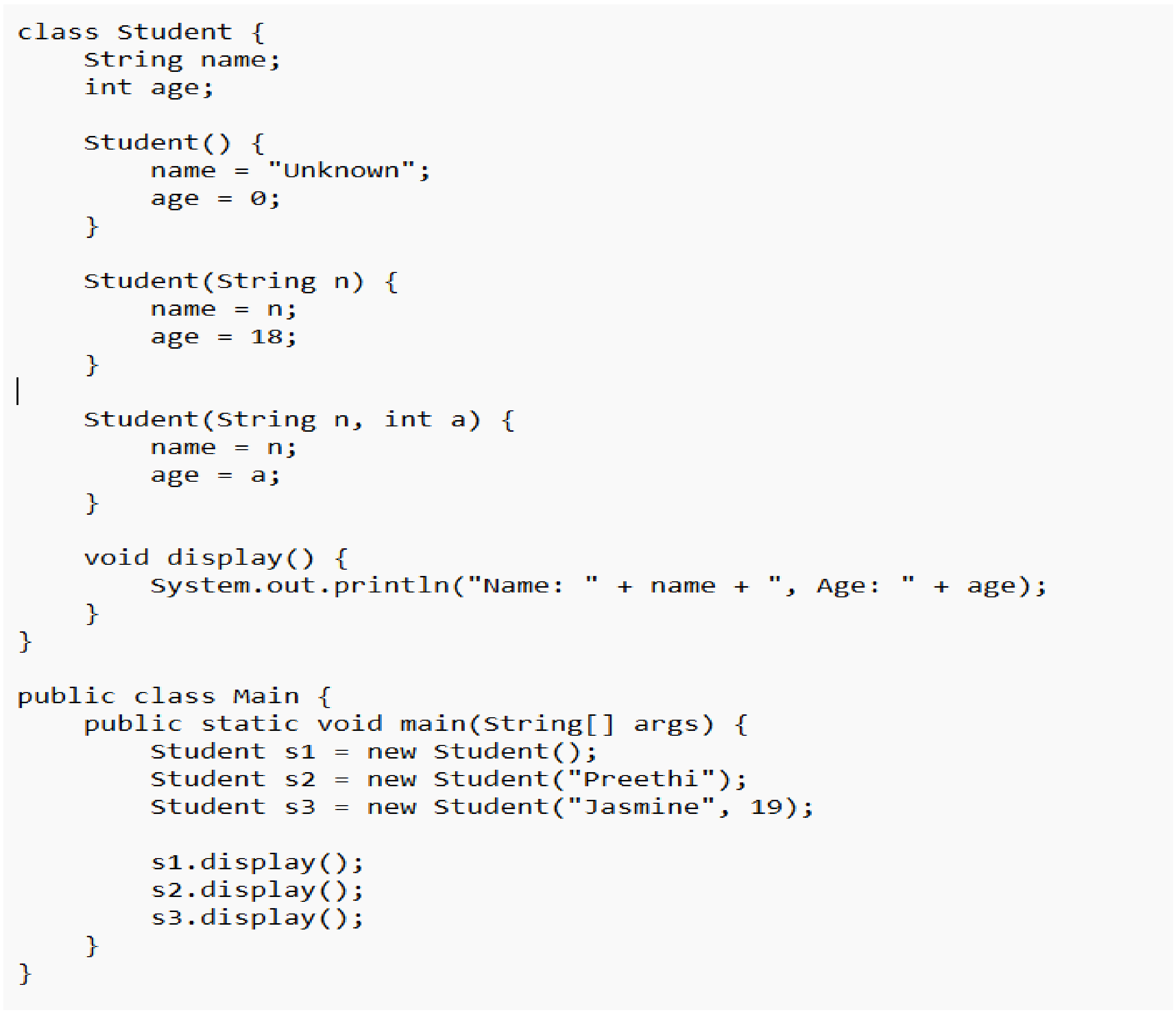
```
C:\Users\PREETHI JASMINE\Desktop>javac Brakes.java

C:\Users\PREETHI JASMINE\Desktop>java Brakes
Uses petrol or diesel
Car engine started
Brakes applied
```

POLYMORPHISM

8) CONSTRUCTOR PROGRAMS

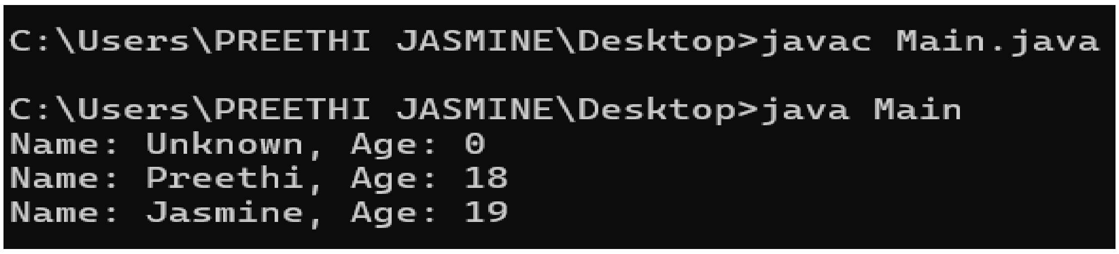
A)Main

CODE:



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

    Student() {
        name = "Unknown";
        age = 0;
    }

    Student(String n) {
        name = n;
        age = 18;
    }

    Student(String n, int a) {
        name = n;
        age = a;
    }

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

public class Main {
    public static void main(String[] args) {
        Student s1 = new Student();
        Student s2 = new Student("Preethi");
        Student s3 = new Student("Jasmine", 19);

        s1.display();
        s2.display();
        s3.display();
    }
}
```

OUTPUT:

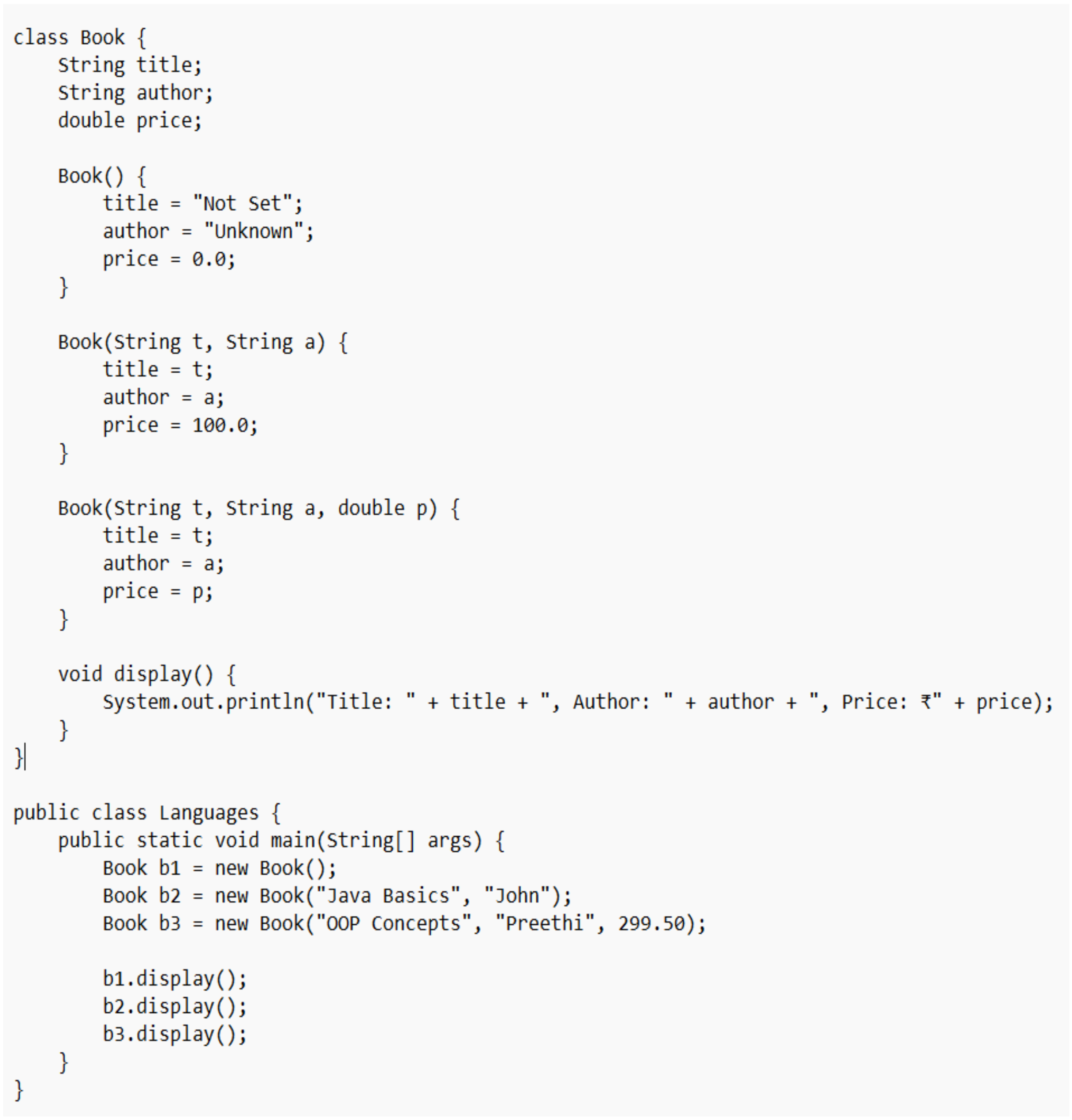


```
C:\Users\PREETHI JASMINE\Desktop>javac Main.java

C:\Users\PREETHI JASMINE\Desktop>java Main
Name: Unknown, Age: 0
Name: Preethi, Age: 18
Name: Jasmine, Age: 19
```

9) CONSTRUCTOR OVERLOADING PROGRAMS

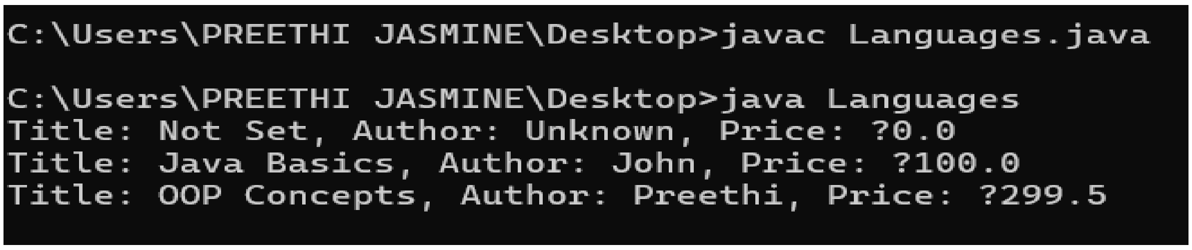
A) Languages

CODE:



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

    Book() {
        title = "Not Set";
        author = "Unknown";
        price = 0.0;
    }

    Book(String t, String a) {
        title = t;
        author = a;
        price = 100.0;
    }

    Book(String t, String a, double p) {
        title = t;
        author = a;
        price = p;
    }

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

public class Languages {
    public static void main(String[] args) {
        Book b1 = new Book();
        Book b2 = new Book("Java Basics", "John");
        Book b3 = new Book("OOP Concepts", "Preethi", 299.50);

        b1.display();
        b2.display();
        b3.display();
    }
}
```

OUTPUT:

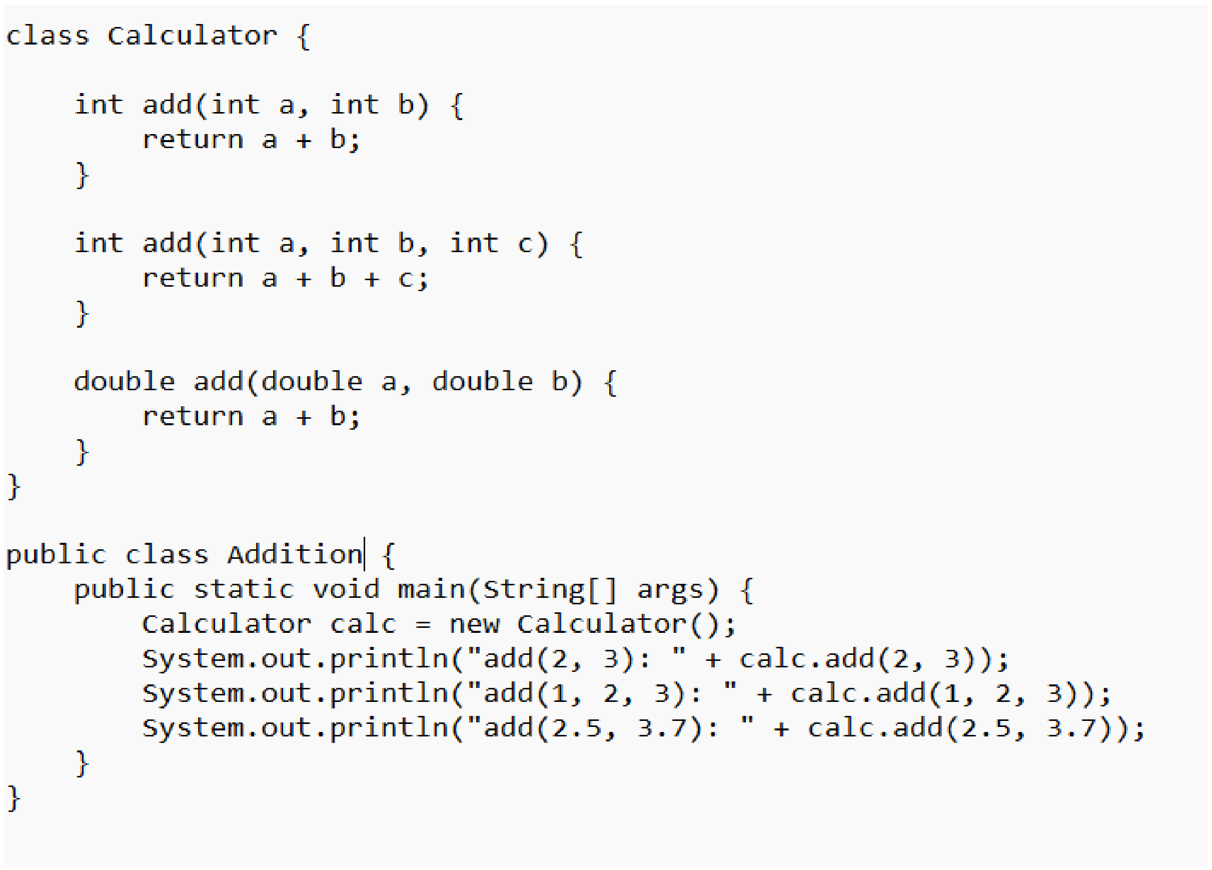


```
C:\Users\PREETHI JASMINE\Desktop>javac Languages.java

C:\Users\PREETHI JASMINE\Desktop>java Languages
Title: Not Set, Author: Unknown, Price: ?0.0
Title: Java Basics, Author: John, Price: ?100.0
Title: OOP Concepts, Author: Preethi, Price: ?299.5
```

10)METHOD OVERLOADING PROGRAMS

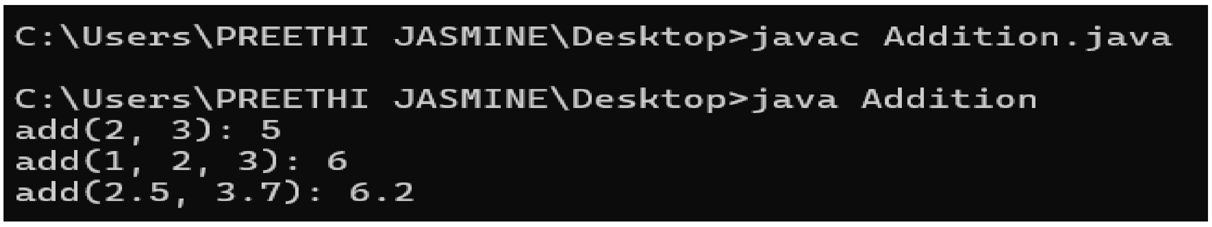
A) Addition

CODE:



```
class Calculator {

    int add(int a, int b) {
        return a + b;
    }

    int add(int a, int b, int c) {
        return a + b + c;
    }

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

public class Addition {
    public static void main(String[] args) {
        Calculator calc = new Calculator();
        System.out.println("add(2, 3): " + calc.add(2, 3));
        System.out.println("add(1, 2, 3): " + calc.add(1, 2, 3));
        System.out.println("add(2.5, 3.7): " + calc.add(2.5, 3.7));
    }
}
```

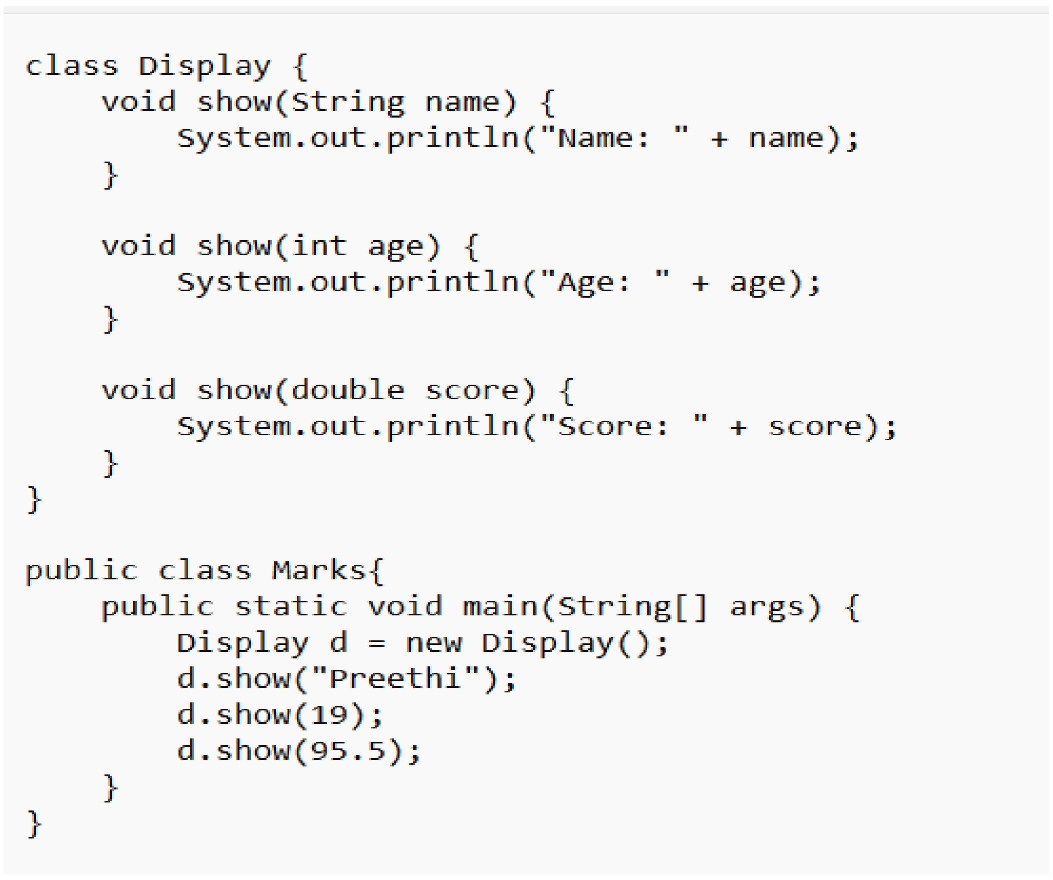
OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac Addition.java

C:\Users\PREETHI JASMINE\Desktop>java Addition
add(2, 3): 5
add(1, 2, 3): 6
add(2.5, 3.7): 6.2
```

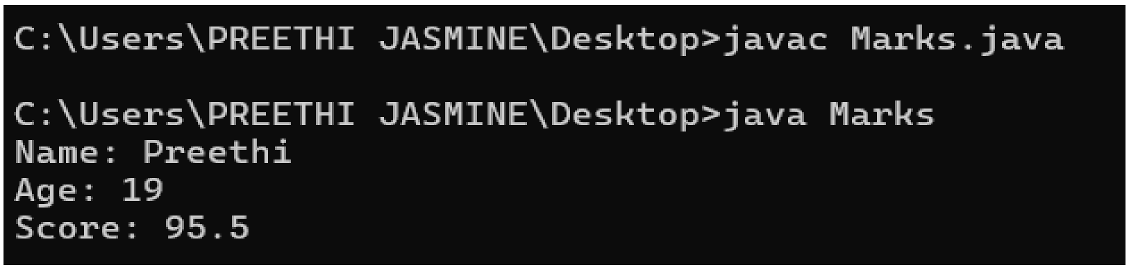
B) Marks

CODE:



```
class Display {
    void show(String name) {
        System.out.println("Name: " + name);
    }

    void show(int age) {
        System.out.println("Age: " + age);
    }

    void show(double score) {
        System.out.println("Score: " + score);
    }
}

public class Marks{
    public static void main(String[] args) {
        Display d = new Display();
        d.show("Preethi");
        d.show(19);
        d.show(95.5);
    }
}
```

OUTPUT:

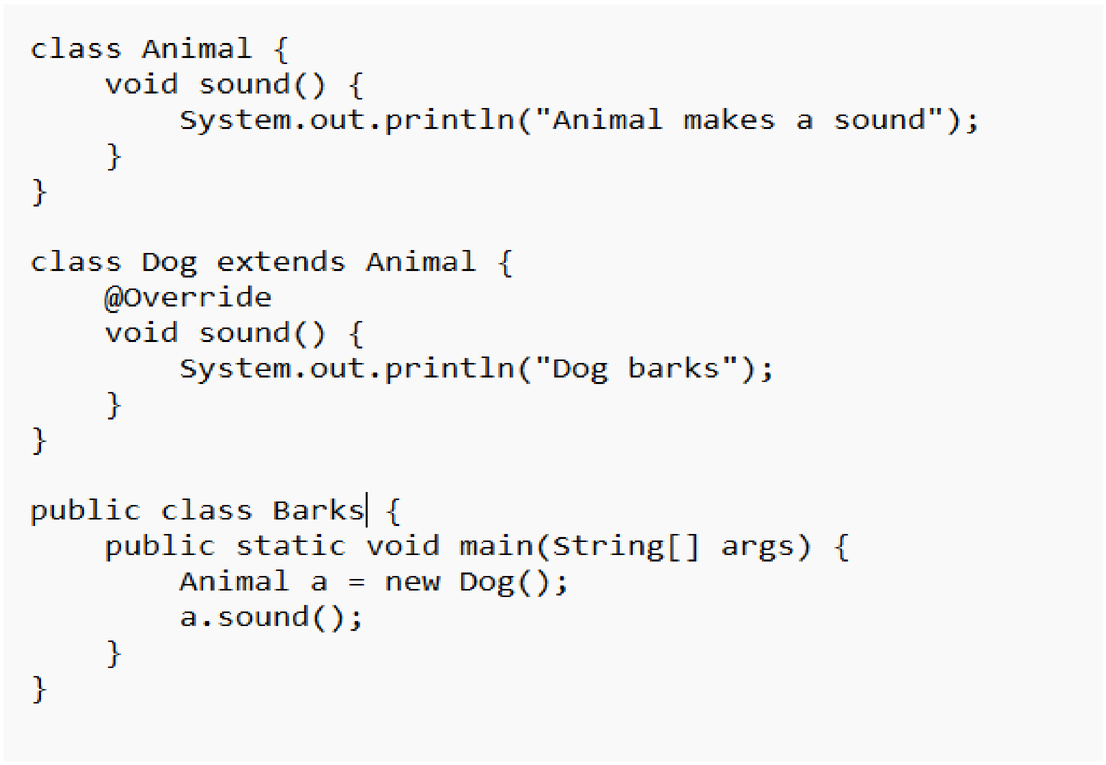


```
C:\Users\PREETHI JASMINE\Desktop>javac Marks.java

C:\Users\PREETHI JASMINE\Desktop>java Marks
Name: Preethi
Age: 19
Score: 95.5
```

11)METHOD OVERRIDDING PROGRAMS

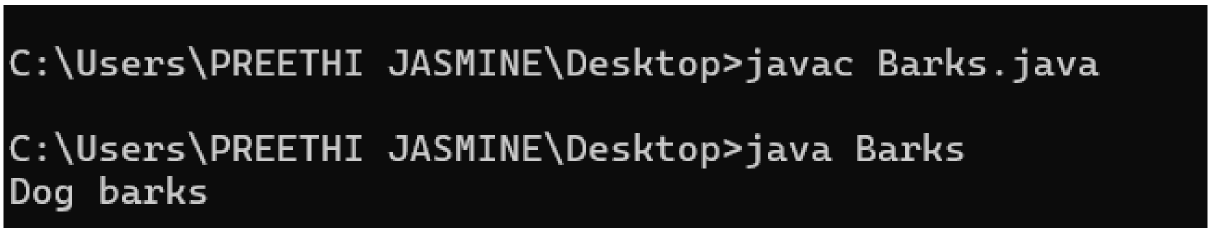
A) Dog Barks

CODE:



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

class Dog extends Animal {
    @Override
    void sound() {
        System.out.println("Dog barks");
    }
}

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

OUTPUT:



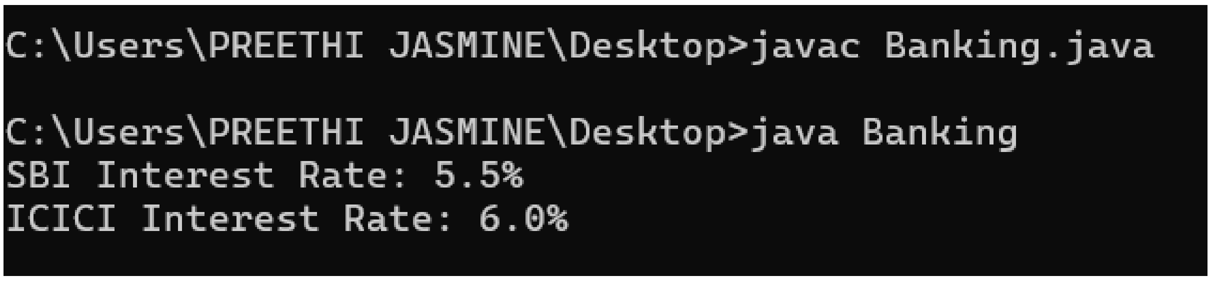
```
C:\Users\PREETHI JASMINE\Desktop>javac Barks.java

C:\Users\PREETHI JASMINE\Desktop>java Barks
Dog barks
```

B) Banking

CODE:

```
class Bank {
    double getInterestRate() {
        return 0;
    }
}

class SBI extends Bank {
    @Override
    double getInterestRate() {
        return 5.5;
    }
}

class ICICI extends Bank {
    @Override
    double getInterestRate() {
        return 6.0;
    }
}

public class Banking{
    public static void main(String[] args) {
        Bank sbi = new SBI();
        Bank icici = new ICICI();

        System.out.println("SBI Interest Rate: " + sbi.getInterestRate() + "%");
        System.out.println("ICICI Interest Rate: " + icici.getInterestRate() + "%");
    }
}
```

OUTPUT:



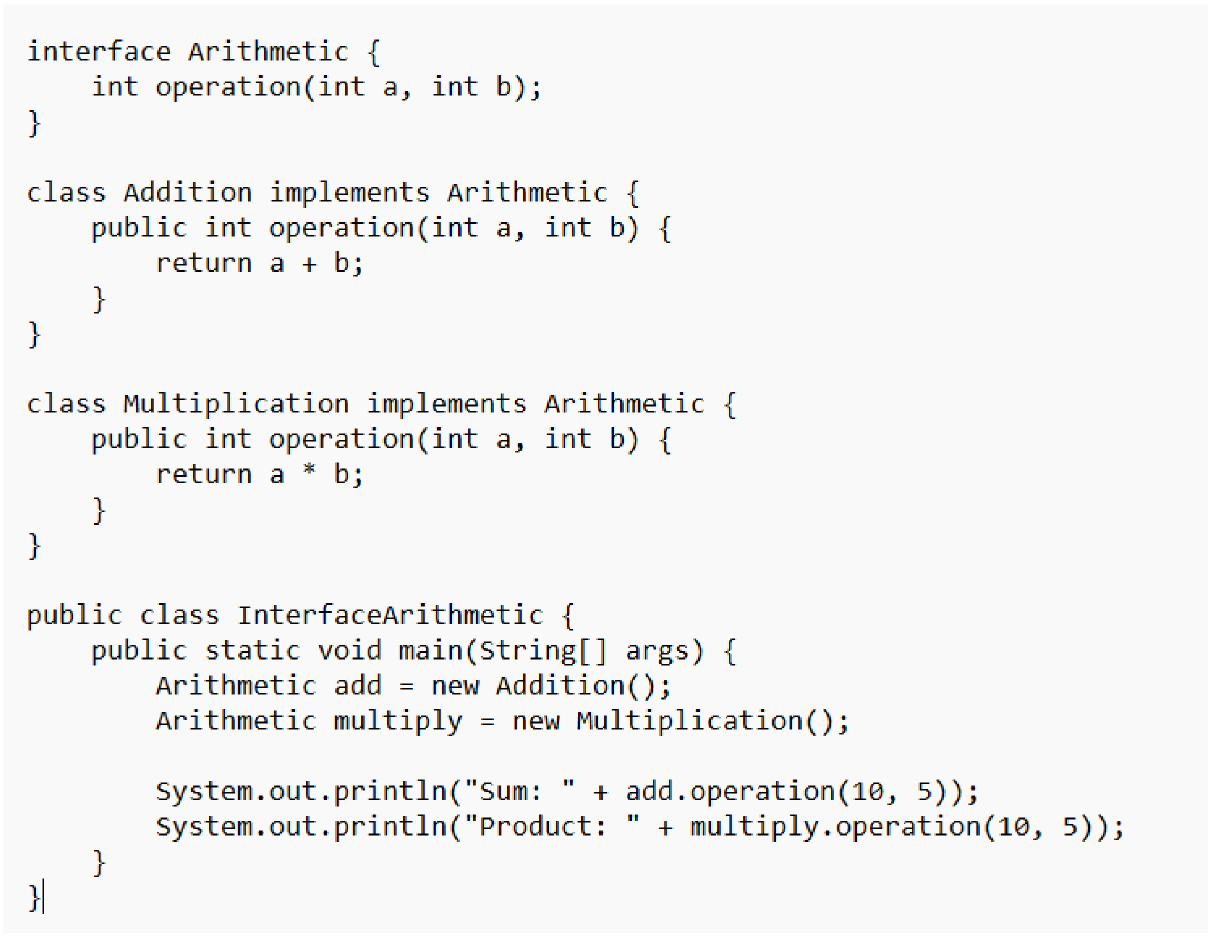
```
C:\Users\PREETHI JASMINE\Desktop>javac Banking.java

C:\Users\PREETHI JASMINE\Desktop>java Banking
SBI Interest Rate: 5.5%
ICICI Interest Rate: 6.0%
```

ABSTRACTION

12)INTERFACE PROGRAMS

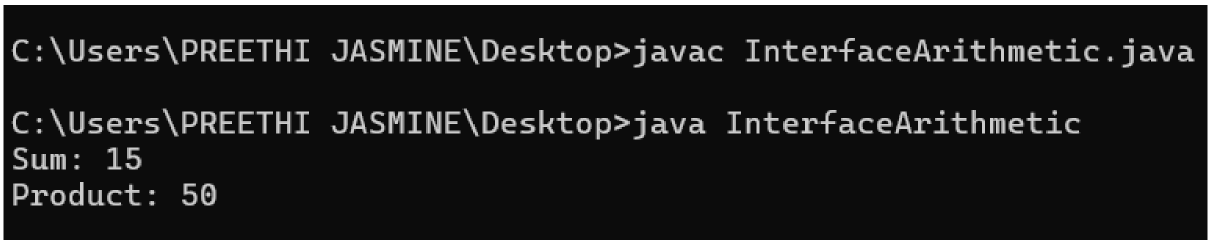
A) Arthimetic operations

CODE:



```
interface Arithmetic {
    int operation(int a, int b);
}

class Addition implements Arithmetic {
    public int operation(int a, int b) {
        return a + b;
    }
}

class Multiplication implements Arithmetic {
    public int operation(int a, int b) {
        return a * b;
    }
}

public class InterfaceArithmetic {
    public static void main(String[] args) {
        Arithmetic add = new Addition();
        Arithmetic multiply = new Multiplication();

        System.out.println("Sum: " + add.operation(10, 5));
        System.out.println("Product: " + multiply.operation(10, 5));
    }
}
```

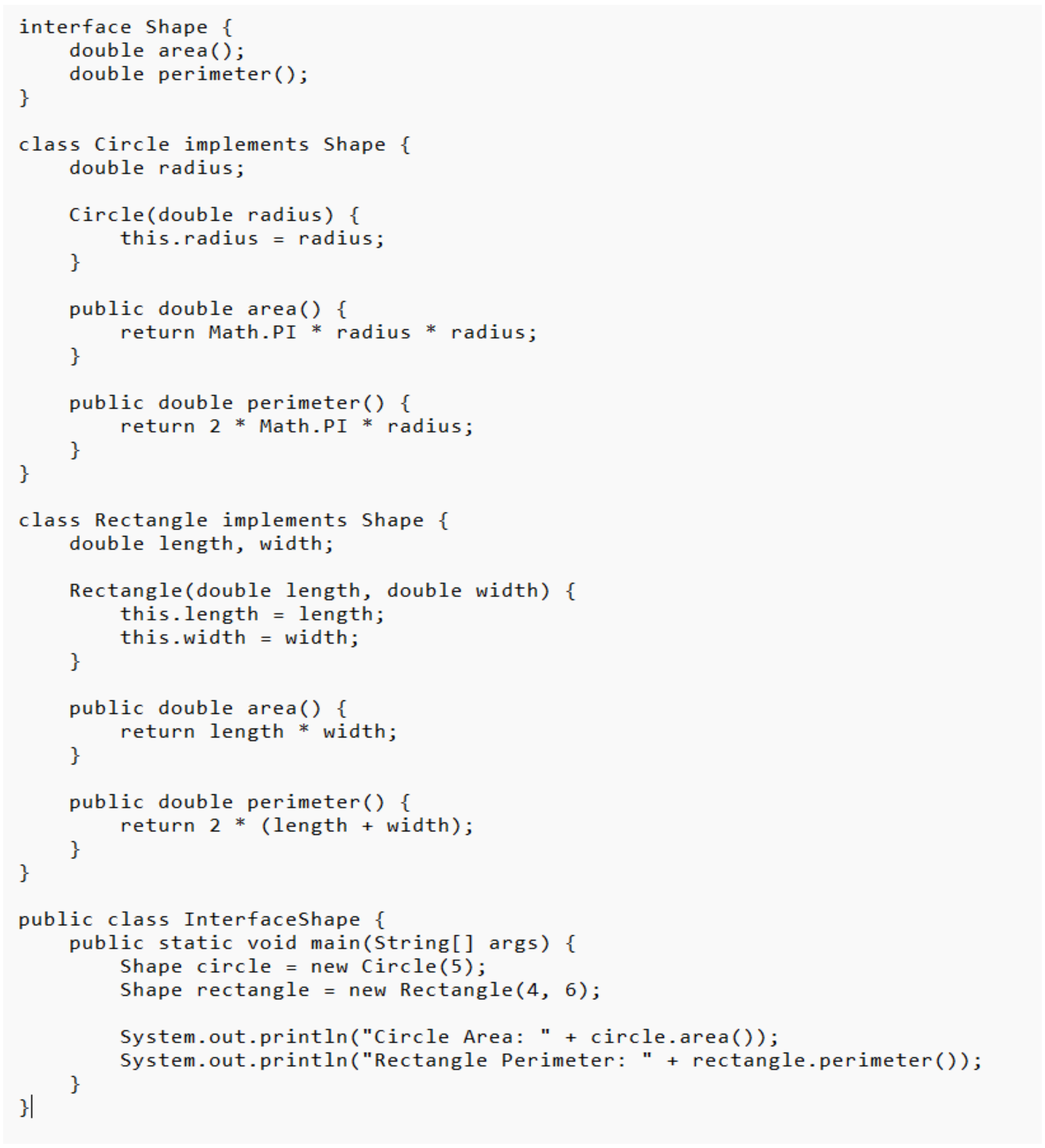
OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac InterfaceArithmetic.java

C:\Users\PREETHI JASMINE\Desktop>java InterfaceArithmetic
Sum: 15
Product: 50
```

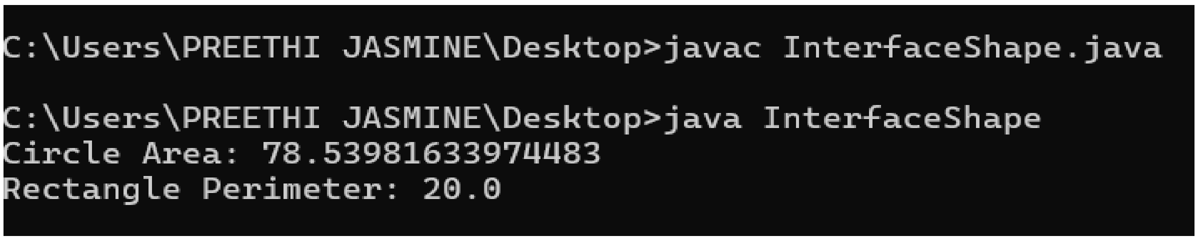
B) Shape Interface

CODE:



```
interface Shape {
    double area();
    double perimeter();
}

class Circle implements Shape {
    double radius;

    Circle(double radius) {
        this.radius = radius;
    }

    public double area() {
        return Math.PI * radius * radius;
    }

    public double perimeter() {
        return 2 * Math.PI * radius;
    }
}

class Rectangle implements Shape {
    double length, width;

    Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }

    public double area() {
        return length * width;
    }

    public double perimeter() {
        return 2 * (length + width);
    }
}

public class InterfaceShape {
    public static void main(String[] args) {
        Shape circle = new Circle(5);
        Shape rectangle = new Rectangle(4, 6);

        System.out.println("Circle Area: " + circle.area());
        System.out.println("Rectangle Perimeter: " + rectangle.perimeter());
    }
}
```

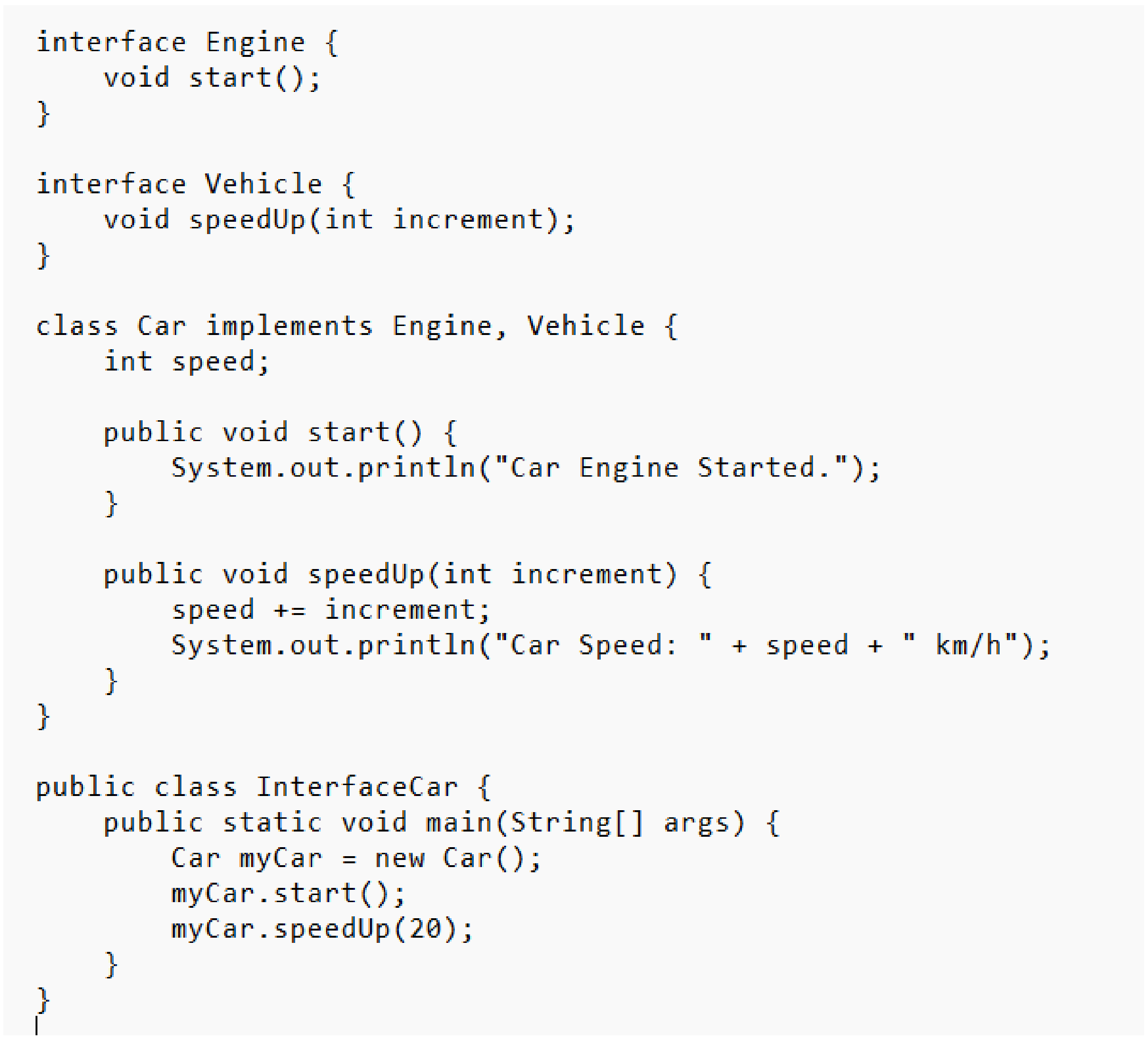
OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac InterfaceShape.java

C:\Users\PREETHI JASMINE\Desktop>java InterfaceShape
Circle Area: 78.53981633974483
Rectangle Perimeter: 20.0
```

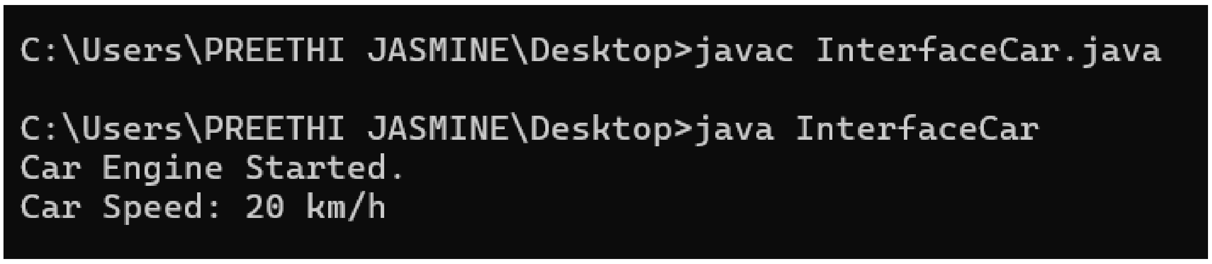
C) Multiple interface

CODE:



```
interface Engine {
    void start();
}

interface Vehicle {
    void speedUp(int increment);
}

class Car implements Engine, Vehicle {
    int speed;

    public void start() {
        System.out.println("Car Engine Started.");
    }

    public void speedUp(int increment) {
        speed += increment;
        System.out.println("Car Speed: " + speed + " km/h");
    }
}

public class InterfaceCar {
    public static void main(String[] args) {
        Car myCar = new Car();
        myCar.start();
        myCar.speedUp(20);
    }
}
```

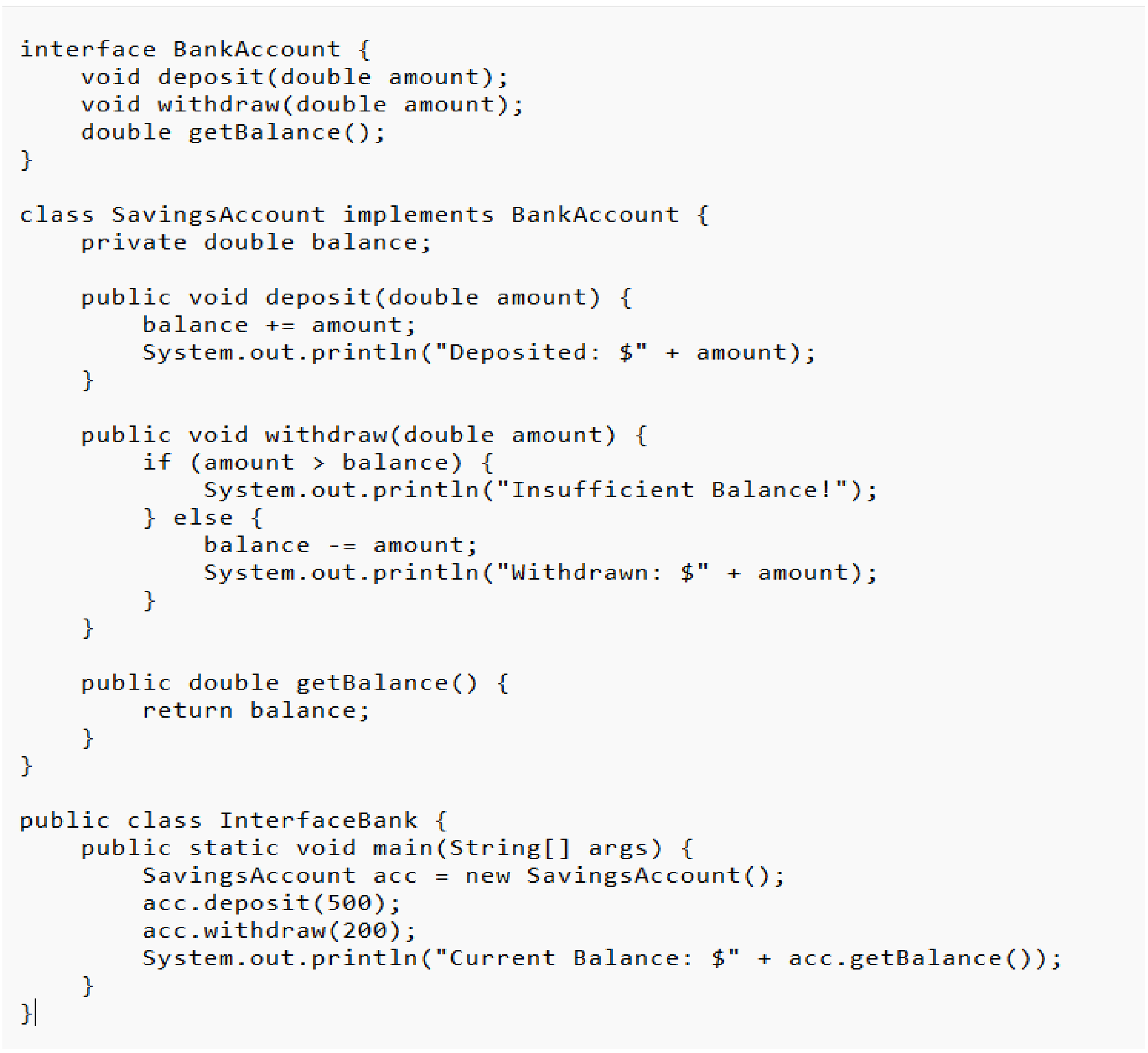
OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac InterfaceCar.java

C:\Users\PREETHI JASMINE\Desktop>java InterfaceCar
Car Engine Started.
Car Speed: 20 km/h
```

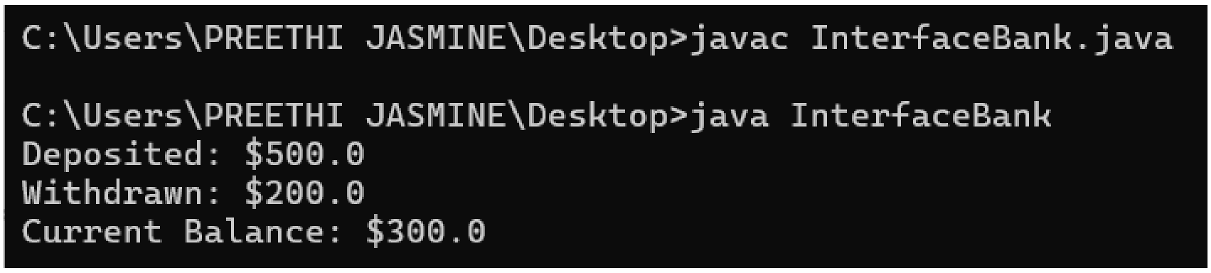
D) Bank transtactions

CODE:



```
interface BankAccount {
    void deposit(double amount);
    void withdraw(double amount);
    double getBalance();
}

class SavingsAccount implements BankAccount {
    private double balance;

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

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

    public double getBalance() {
        return balance;
    }
}

public class InterfaceBank {
    public static void main(String[] args) {
        SavingsAccount acc = new SavingsAccount();
        acc.deposit(500);
        acc.withdraw(200);
        System.out.println("Current Balance: $" + acc.getBalance());
    }
}
```

OUTPUT:

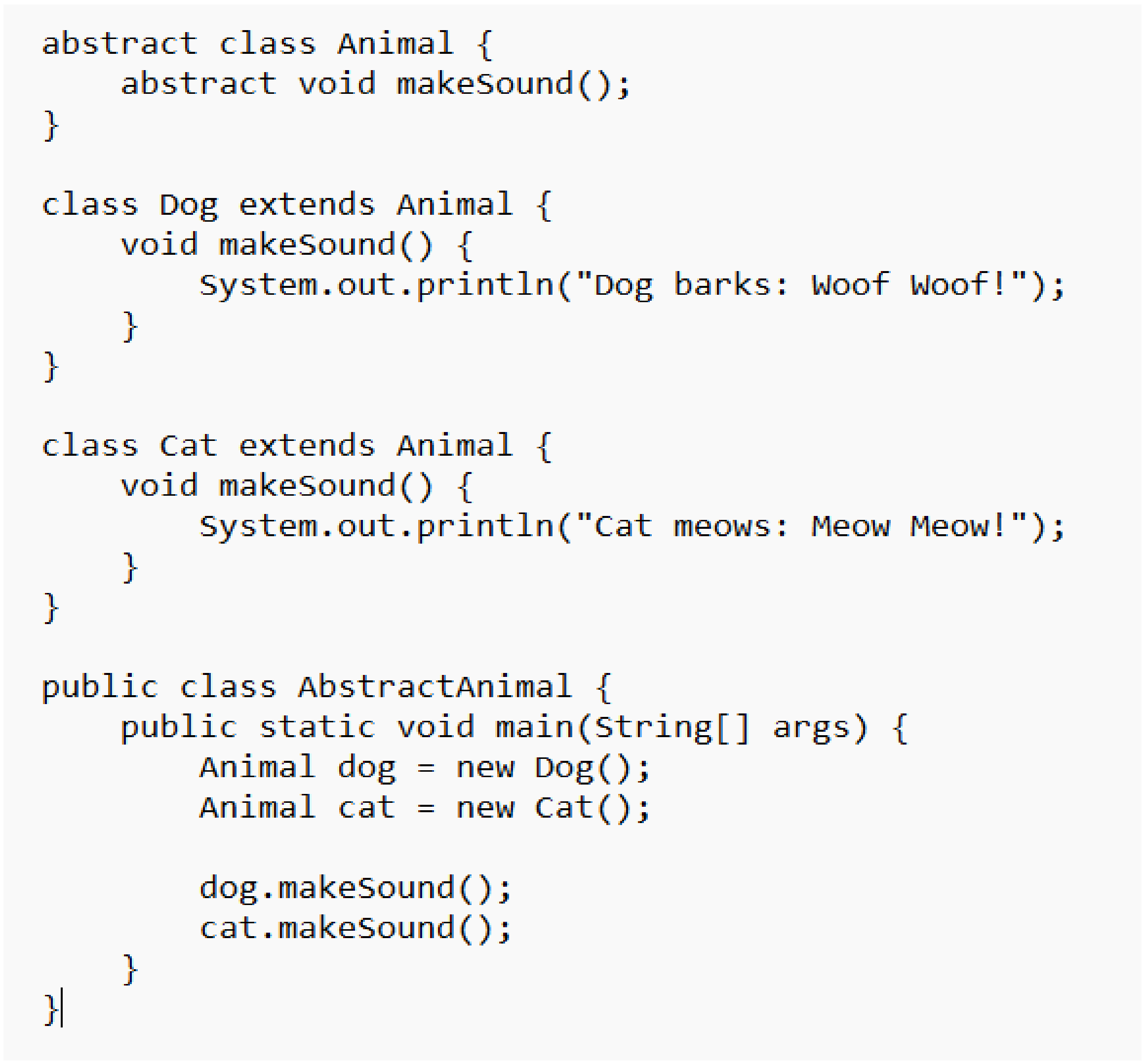


```
C:\Users\PREETHI JASMINE\Desktop>javac InterfaceBank.java

C:\Users\PREETHI JASMINE\Desktop>java InterfaceBank
Deposited: $500.0
Withdrawn: $200.0
Current Balance: $300.0
```

13)ABSTRACT CLASS PROGRAMS

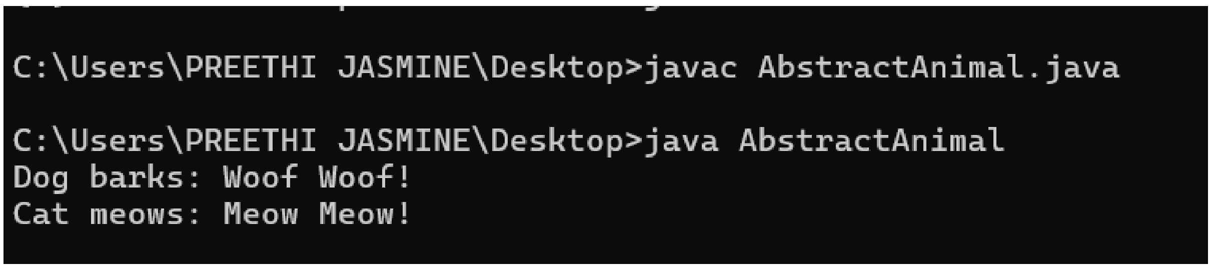
A) Animal Sounds

CODE:



```
abstract class Animal {
    abstract void makeSound();
}

class Dog extends Animal {
    void makeSound() {
        System.out.println("Dog barks: Woof Woof!");
    }
}

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

public class AbstractAnimal {
    public static void main(String[] args) {
        Animal dog = new Dog();
        Animal cat = new Cat();

        dog.makeSound();
        cat.makeSound();
    }
}
```

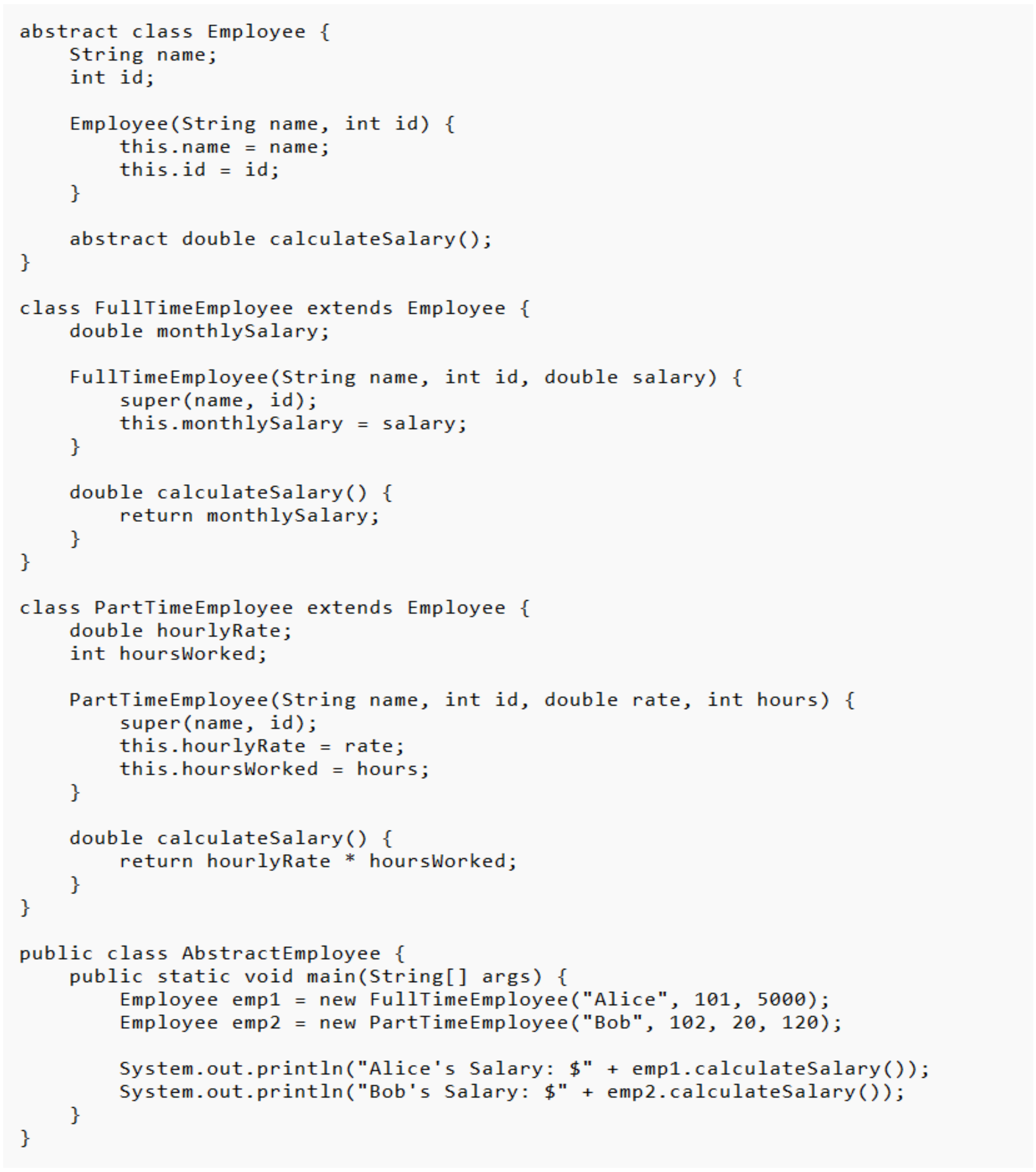
OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac AbstractAnimal.java

C:\Users\PREETHI JASMINE\Desktop>java AbstractAnimal
Dog barks: Woof Woof!
Cat meows: Meow Meow!
```

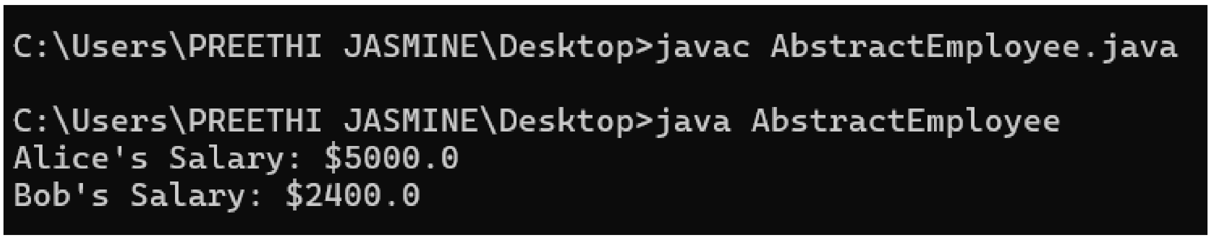
B) Employee Salary Calculation

CODE:



```java
abstract class Employee {
    String name;
    int id;

    Employee(String name, int id) {
        this.name = name;
        this.id = id;
    }

    abstract double calculateSalary();
}

class FullTimeEmployee extends Employee {
    double monthlySalary;

    FullTimeEmployee(String name, int id, double salary) {
        super(name, id);
        this.monthlySalary = salary;
    }

    double calculateSalary() {
        return monthlySalary;
    }
}

class PartTimeEmployee extends Employee {
    double hourlyRate;
    int hoursWorked;

    PartTimeEmployee(String name, int id, double rate, int hours) {
        super(name, id);
        this.hourlyRate = rate;
        this.hoursWorked = hours;
    }

    double calculateSalary() {
        return hourlyRate * hoursWorked;
    }
}

public class AbstractEmployee {
    public static void main(String[] args) {
        Employee emp1 = new FullTimeEmployee("Alice", 101, 5000);
        Employee emp2 = new PartTimeEmployee("Bob", 102, 20, 120);

        System.out.println("Alice's Salary: $" + emp1.calculateSalary());
        System.out.println("Bob's Salary: $" + emp2.calculateSalary());
    }
}
```

OUTPUT:



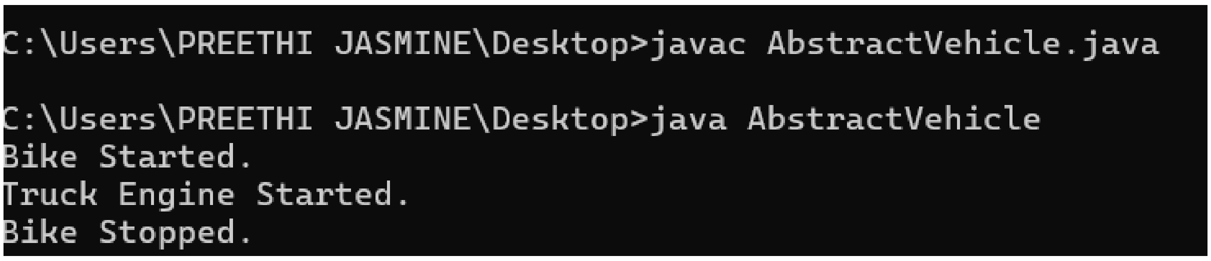
```
C:\Users\PREETHI JASMINE\Desktop>javac AbstractEmployee.java

C:\Users\PREETHI JASMINE\Desktop>java AbstractEmployee
Alice's Salary: $5000.0
Bob's Salary: $2400.0
```

C) Vehicle Features

CODE:

```
abstract class Vehicle {
    abstract void start();
    abstract void stop();
}

class Bike extends Vehicle {
    void start() {
        System.out.println("Bike Started.");
    }

    void stop() {
        System.out.println("Bike Stopped.");
    }
}

class Truck extends Vehicle {
    void start() {
        System.out.println("Truck Engine Started.");
    }

    void stop() {
        System.out.println("Truck Stopped.");
    }
}

public class AbstractVehicle {
    public static void main(String[] args) {
        Vehicle bike = new Bike();
        Vehicle truck = new Truck();

        bike.start();
        truck.start();
        bike.stop();
    }
}
```

OUTPUT:



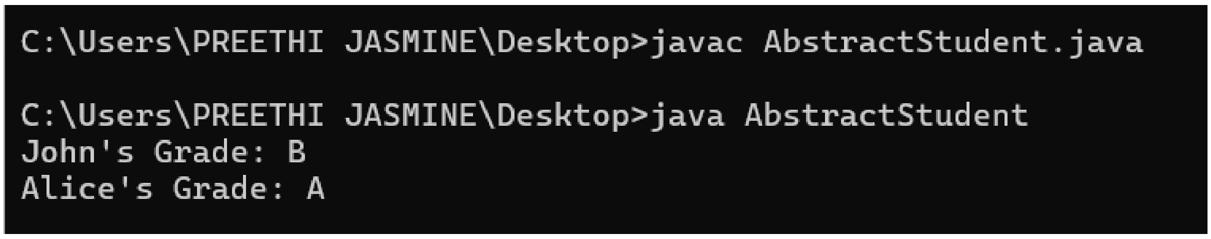
```
C:\Users\PREETHI JASMINE\Desktop>javac AbstractVehicle.java

C:\Users\PREETHI JASMINE\Desktop>java AbstractVehicle
Bike Started.
Truck Engine Started.
Bike Stopped.
```

D) Student Result Calculation

CODE:

```java
abstract class Student {
    String name;
    int rollNumber;

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

    abstract void calculateGrade();
}

class EngineeringStudent extends Student {
    double marks;

    EngineeringStudent(String name, int rollNumber, double marks) {
        super(name, rollNumber);
        this.marks = marks;
    }

    void calculateGrade() {
        if (marks >= 90) {
            System.out.println(name + "'s Grade: A");
        } else if (marks >= 75) {
            System.out.println(name + "'s Grade: B");
        } else {
            System.out.println(name + "'s Grade: C");
        }
    }
}

class MedicalStudent extends Student {
    double marks;

    MedicalStudent(String name, int rollNumber, double marks) {
        super(name, rollNumber);
        this.marks = marks;
    }

    void calculateGrade() {
        if (marks >= 85) {
            System.out.println(name + "'s Grade: A");
        } else if (marks >= 70) {
            System.out.println(name + "'s Grade: B");
        } else {
            System.out.println(name + "'s Grade: C");
        }
    }
}

public class AbstractStudent {
    public static void main(String[] args) {
        Student enggStudent = new EngineeringStudent("John", 201, 88);
        Student medStudent = new MedicalStudent("Alice", 301, 92);

        enggStudent.calculateGrade();
        medStudent.calculateGrade();
    }
}
```

OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac AbstractStudent.java

C:\Users\PREETHI JASMINE\Desktop>java AbstractStudent
John's Grade: B
Alice's Grade: A
```

ENCAPSULATION

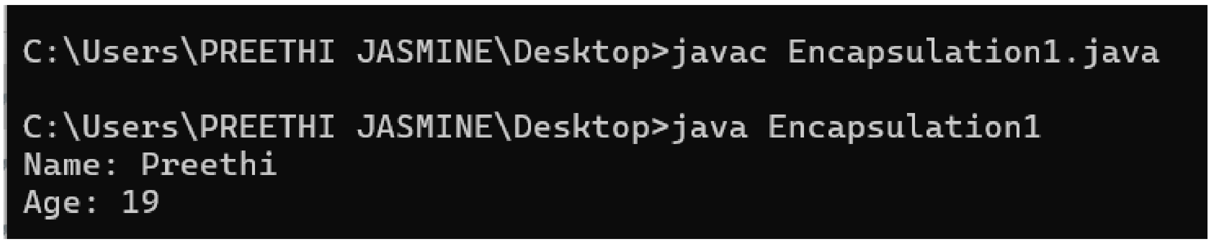
14) Person Details

A)

CODE:

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

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        if(age > 0) {
            this.age = age;
        }
    }
}

public class Encapsulation1 {
    public static void main(String[] args) {
        Person p = new Person();
        p.setName("Preethi");
        p.setAge(19);

        System.out.println("Name: " + p.getName());
        System.out.println("Age: " + p.getAge());
    }
}
```

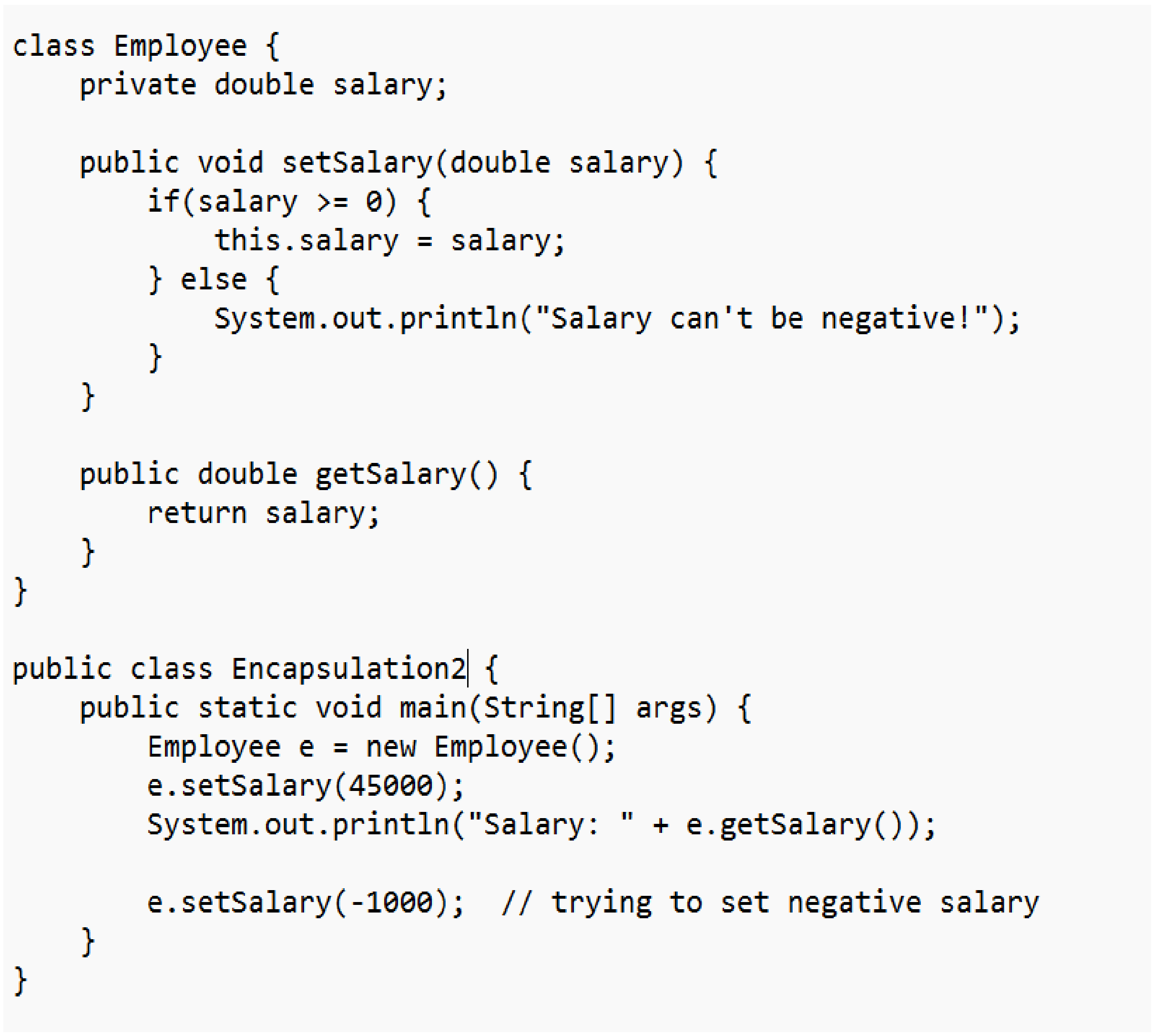
OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac Encapsulation1.java

C:\Users\PREETHI JASMINE\Desktop>java Encapsulation1
Name: Preethi
Age: 19
```

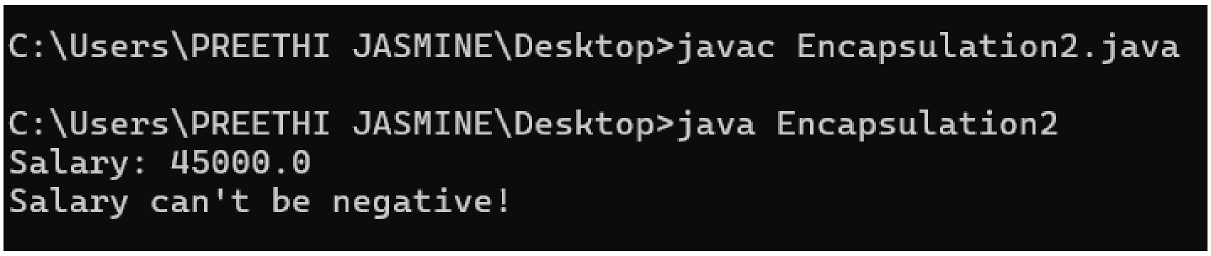
B) Employee Validation

CODE:



```java
class Employee {
    private double salary;

    public void setSalary(double salary) {
        if(salary >= 0) {
            this.salary = salary;
        } else {
            System.out.println("Salary can't be negative!");
        }
    }

    public double getSalary() {
        return salary;
    }
}

public class Encapsulation2 {
    public static void main(String[] args) {
        Employee e = new Employee();
        e.setSalary(45000);
        System.out.println("Salary: " + e.getSalary());

        e.setSalary(-1000);  // trying to set negative salary
    }
}
```

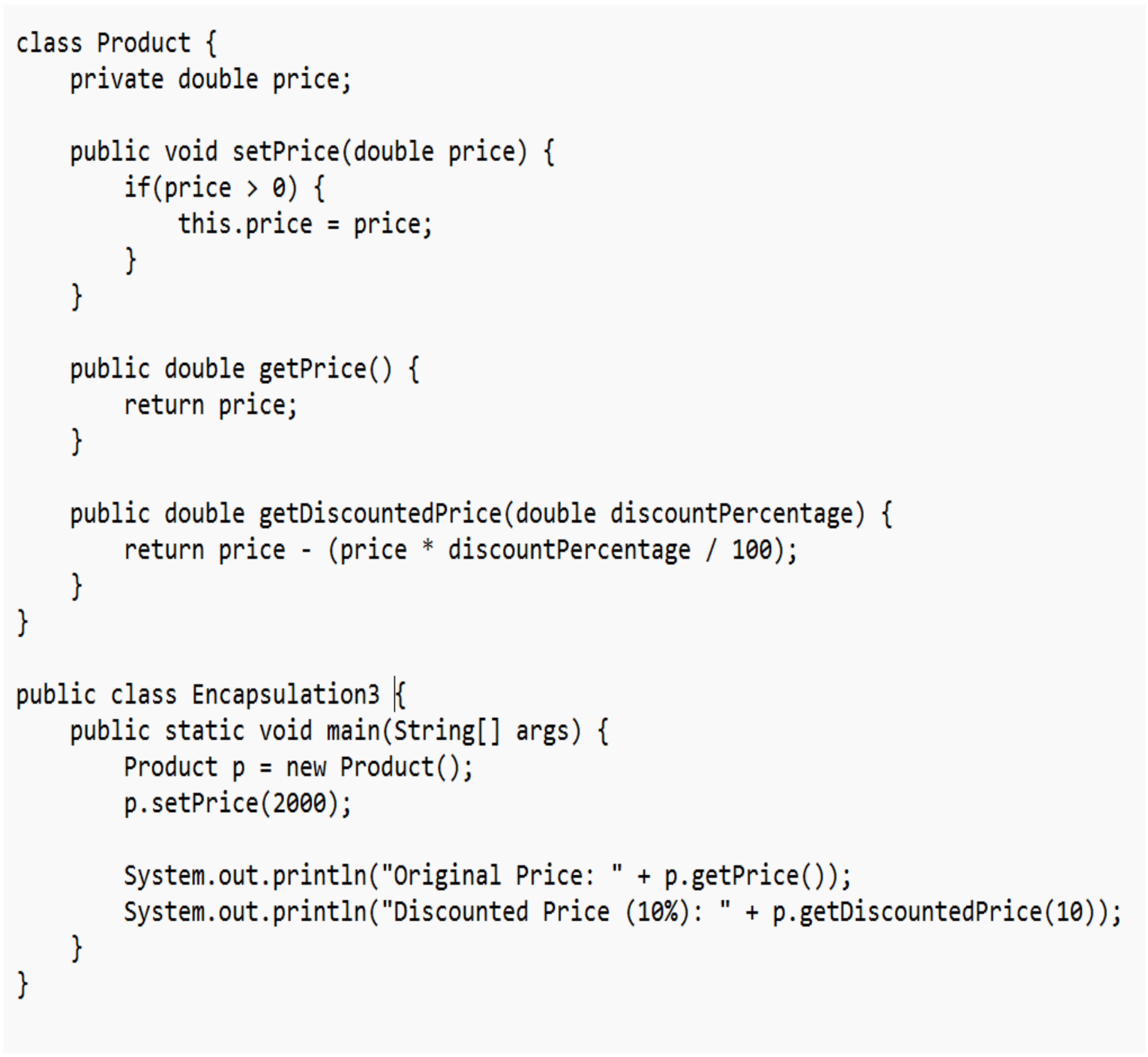
OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac Encapsulation2.java

C:\Users\PREETHI JASMINE\Desktop>java Encapsulation2
Salary: 45000.0
Salary can't be negative!
```

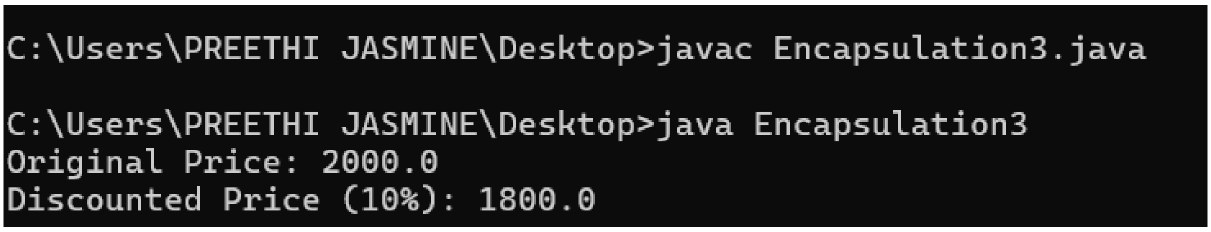
C) Product Discount

CODE:



```java
class Product {
    private double price;

    public void setPrice(double price) {
        if(price > 0) {
            this.price = price;
        }
    }

    public double getPrice() {
        return price;
    }

    public double getDiscountedPrice(double discountPercentage) {
        return price - (price * discountPercentage / 100);
    }
}

public class Encapsulation3 {
    public static void main(String[] args) {
        Product p = new Product();
        p.setPrice(2000);

        System.out.println("Original Price: " + p.getPrice());
        System.out.println("Discounted Price (10%): " + p.getDiscountedPrice(10));
    }
}
```

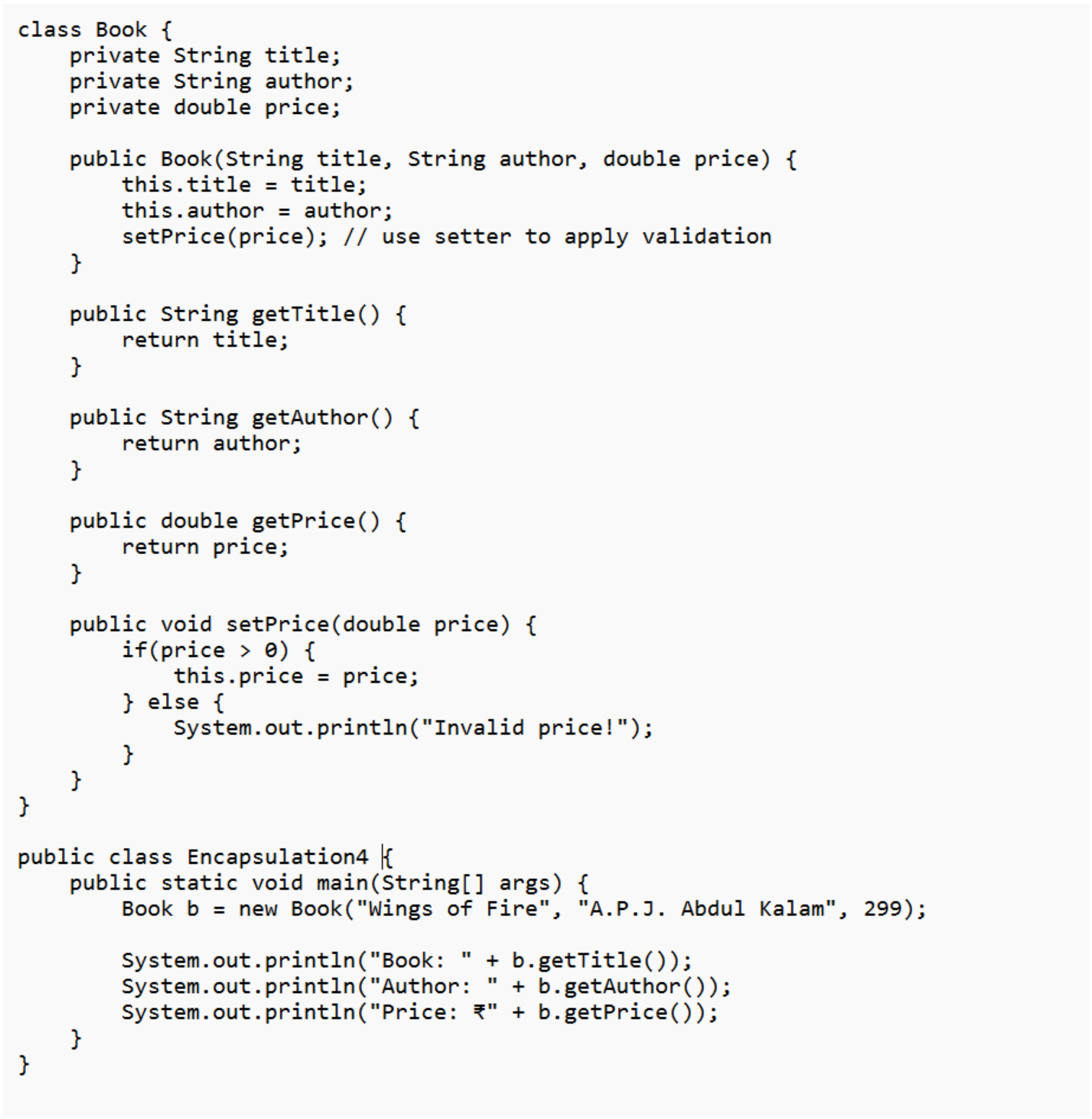
OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac Encapsulation3.java

C:\Users\PREETHI JASMINE\Desktop>java Encapsulation3
Original Price: 2000.0
Discounted Price (10%): 1800.0
```

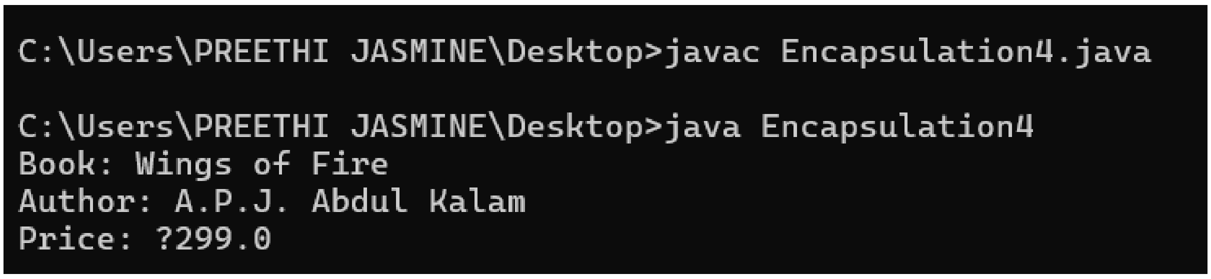
D) Books Price

CODE:



```
class Book {
    private String title;
    private String author;
    private double price;

    public Book(String title, String author, double price) {
        this.title = title;
        this.author = author;
        setPrice(price); // use setter to apply validation
    }

    public String getTitle() {
        return title;
    }

    public String getAuthor() {
        return author;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        if(price > 0) {
            this.price = price;
        } else {
            System.out.println("Invalid price!");
        }
    }
}

public class Encapsulation4 {
    public static void main(String[] args) {
        Book b = new Book("Wings of Fire", "A.P.J. Abdul Kalam", 299);

        System.out.println("Book: " + b.getTitle());
        System.out.println("Author: " + b.getAuthor());
        System.out.println("Price: ₹" + b.getPrice());
    }
}
```

OUTPUT:



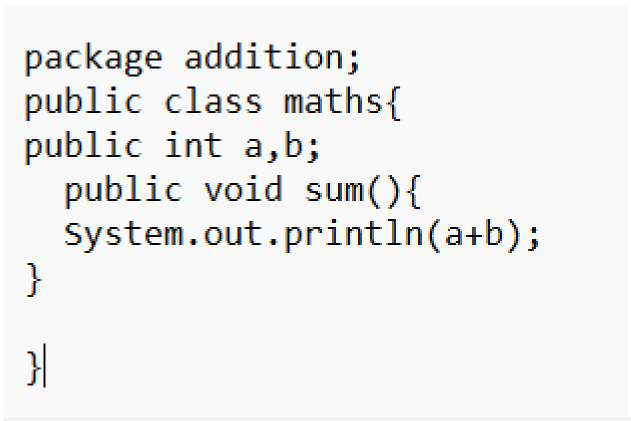
```
C:\Users\PREETHI JASMINE\Desktop>javac Encapsulation4.java

C:\Users\PREETHI JASMINE\Desktop>java Encapsulation4
Book: Wings of Fire
Author: A.P.J. Abdul Kalam
Price: ?299.0
```

PACKAGES

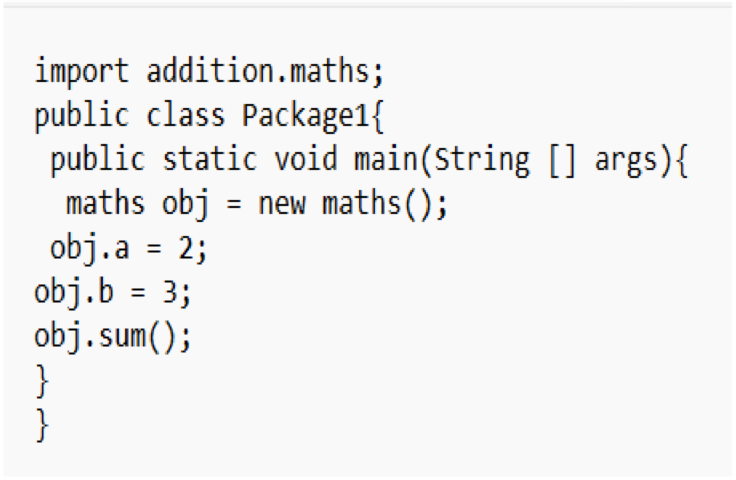
15) User Defined Packages

A) Maths

CODE:

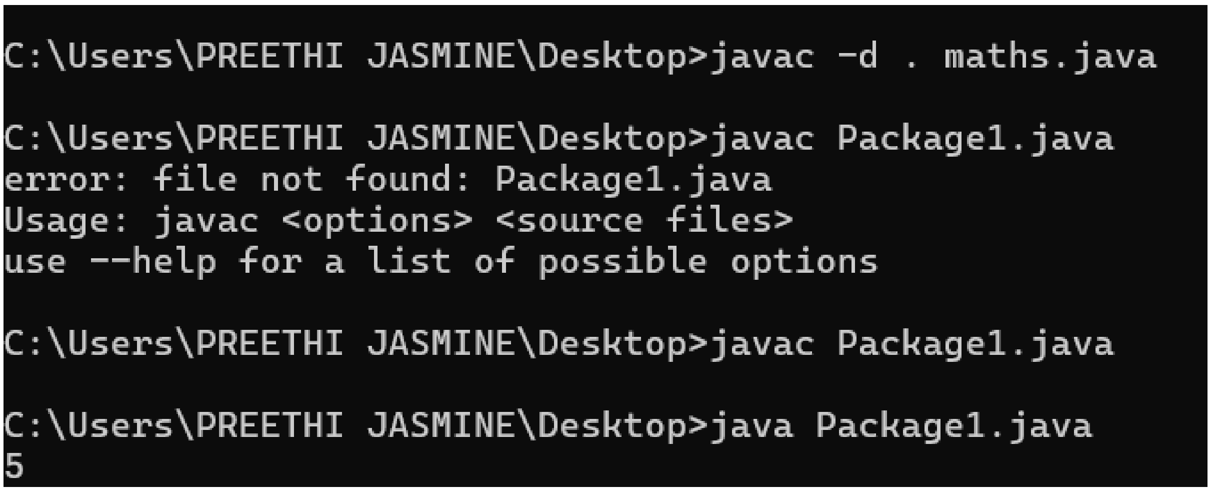


```
package addition;
public class maths{
public int a,b;
  public void sum(){
  System.out.println(a+b);
}

}
```



```
 import addition.maths;
 public class Package1{
  public static void main(String [] args){
   maths obj = new maths();
  obj.a = 2;
 obj.b = 3;
 obj.sum();
 }
 }
```

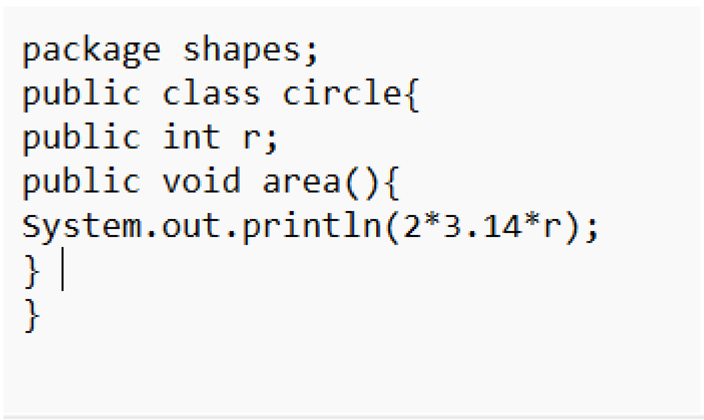
OUTPUT:



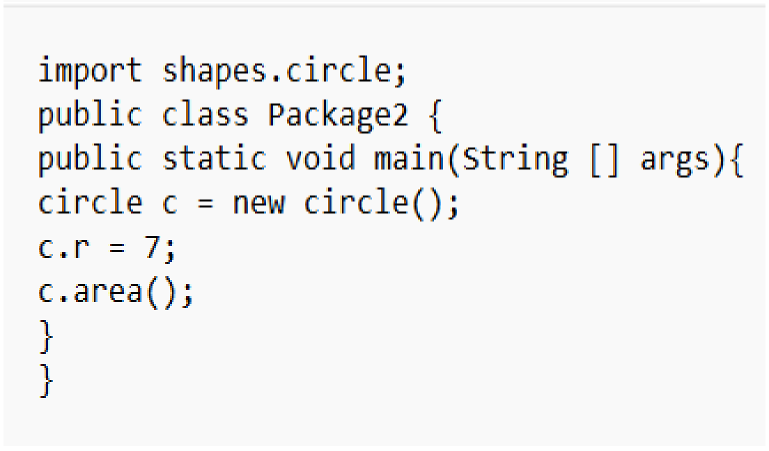
```
C:\Users\PREETHI JASMINE\Desktop>javac -d . maths.java

C:\Users\PREETHI JASMINE\Desktop>javac Package1.java
error: file not found: Package1.java
Usage: javac <options> <source files>
use --help for a list of possible options

C:\Users\PREETHI JASMINE\Desktop>javac Package1.java

C:\Users\PREETHI JASMINE\Desktop>java Package1.java
5
```

B) Shapes

CODE:

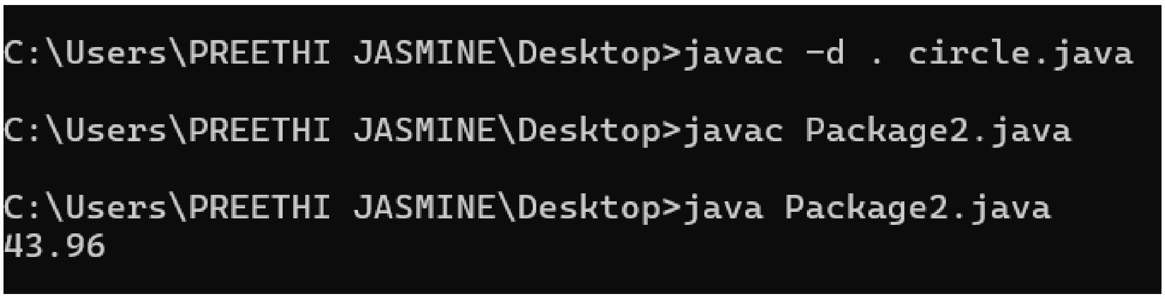


```
package shapes;
public class circle{
public int r;
public void area(){
System.out.println(2*3.14*r);
}
}
```



```
import shapes.circle;
public class Package2 {
public static void main(String [] args){
circle c = new circle();
c.r = 7;
c.area();
}
}
```

OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac -d . circle.java

C:\Users\PREETHI JASMINE\Desktop>javac Package2.java

C:\Users\PREETHI JASMINE\Desktop>java Package2.java
43.96
```

15)Build In Programs

A)Maths Operations

CODE:

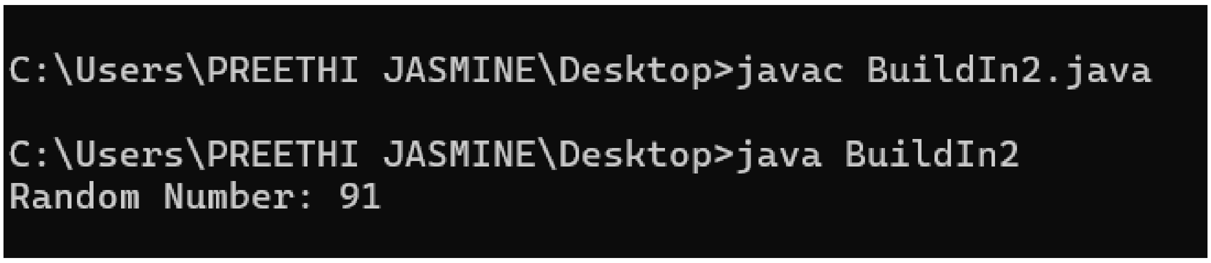
```
public class BuildIn1 {
    public static void main(String[] args) {
        try {
            int a = 10, b = 0;
            int result = a / b; // This will cause ArithmeticException
            System.out.println("Result: " + result);
        } catch (ArithmeticException e) {
            System.out.println("Error: Cannot divide by zero.");
        }
        System.out.println("Program continues...");
    }
}
```

B)Random number

CODE:

```
import java.util.Random;
public class BuildIn2 {
public static void main(String[] args) {
Random rand = new Random();
System.out.println("Random Number: " +
rand.nextInt(100));
}
}
```

OUTPUT:

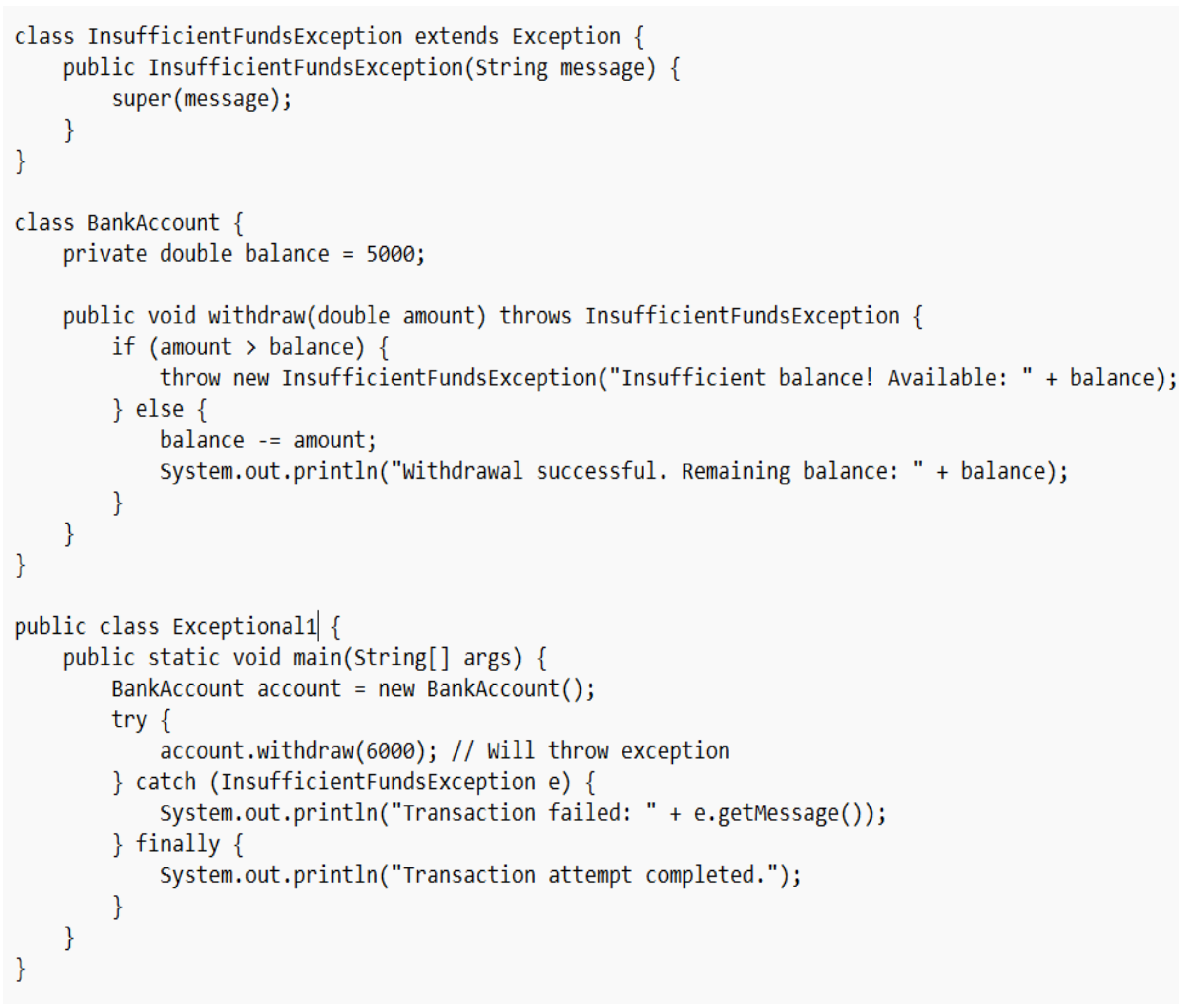


```
C:\Users\PREETHI JASMINE\Desktop>javac BuildIn2.java

C:\Users\PREETHI JASMINE\Desktop>java BuildIn2
Random Number: 91
```

16) EXCEPTIONAL HANDLING

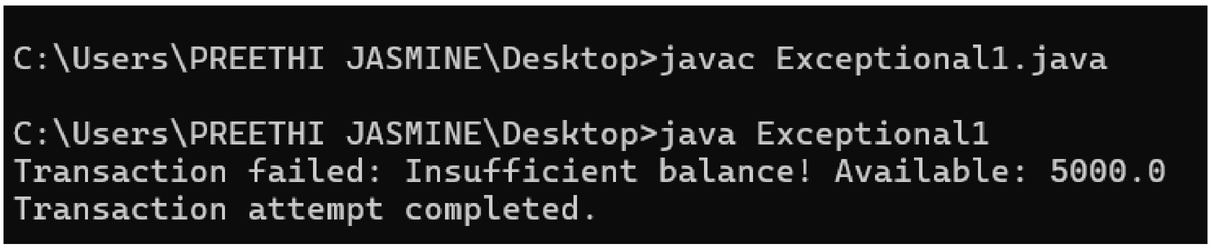
A) Banking App

CODE:



```java
class InsufficientFundsException extends Exception {
    public InsufficientFundsException(String message) {
        super(message);
    }
}

class BankAccount {
    private double balance = 5000;

    public void withdraw(double amount) throws InsufficientFundsException {
        if (amount > balance) {
            throw new InsufficientFundsException("Insufficient balance! Available: " + balance);
        } else {
            balance -= amount;
            System.out.println("Withdrawal successful. Remaining balance: " + balance);
        }
    }
}

public class Exceptional1 {
    public static void main(String[] args) {
        BankAccount account = new BankAccount();
        try {
            account.withdraw(6000); // Will throw exception
        } catch (InsufficientFundsException e) {
            System.out.println("Transaction failed: " + e.getMessage());
        } finally {
            System.out.println("Transaction attempt completed.");
        }
    }
}
```

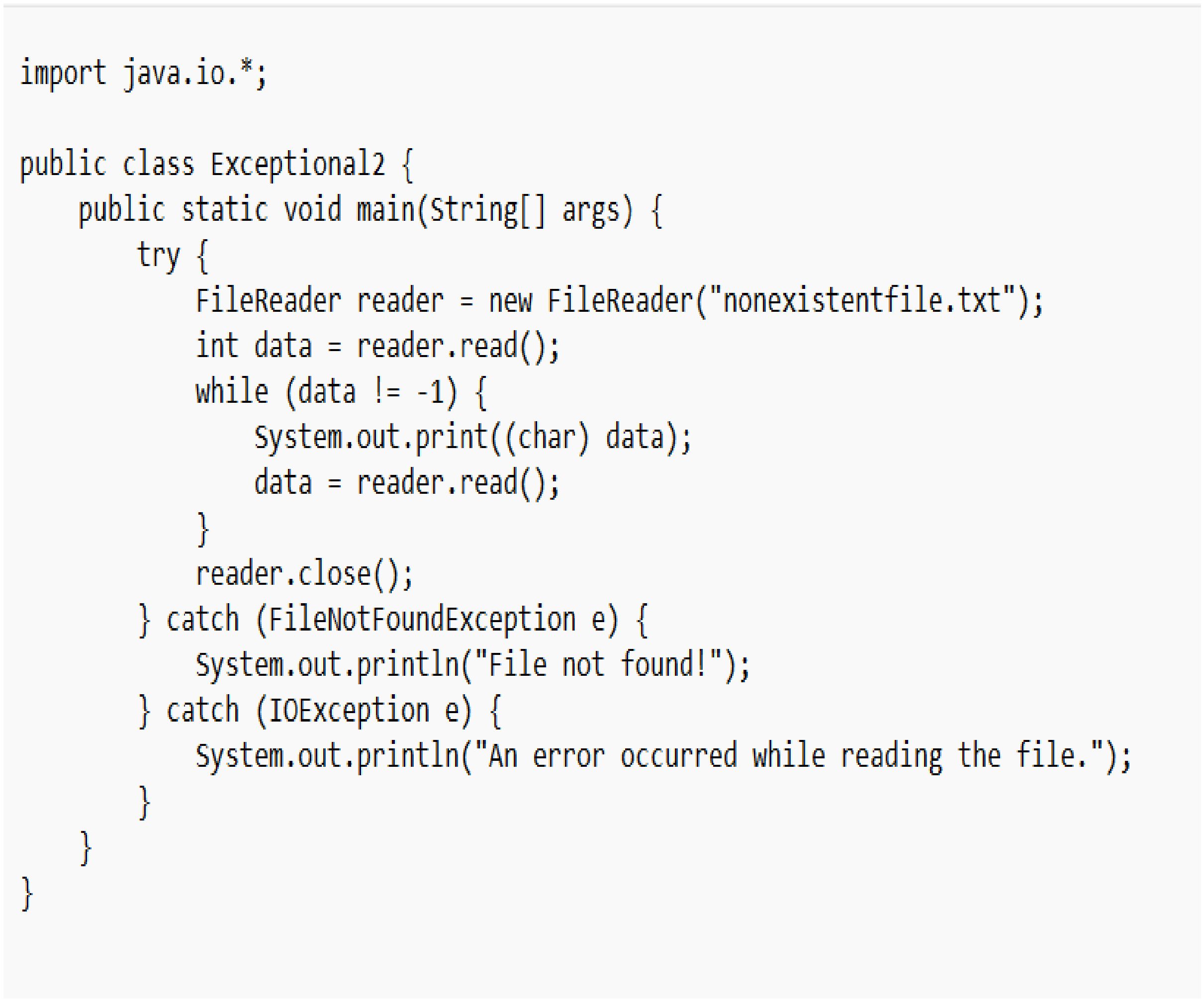
OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac Exceptional1.java

C:\Users\PREETHI JASMINE\Desktop>java Exceptional1
Transaction failed: Insufficient balance! Available: 5000.0
Transaction attempt completed.
```

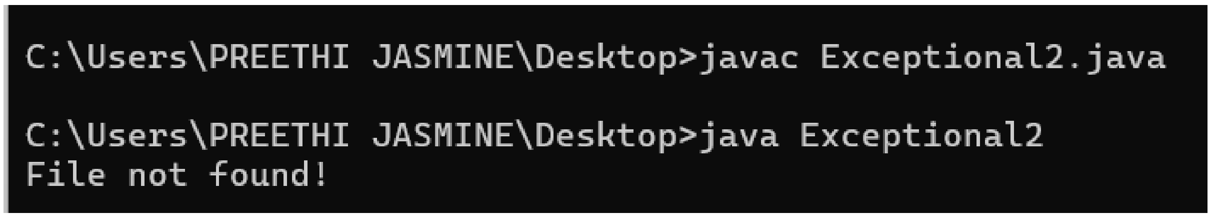
B) File Example

CODE:



```java
import java.io.*;

public class Exceptional2 {
    public static void main(String[] args) {
        try {
            FileReader reader = new FileReader("nonexistentfile.txt");
            int data = reader.read();
            while (data != -1) {
                System.out.print((char) data);
                data = reader.read();
            }
            reader.close();
        } catch (FileNotFoundException e) {
            System.out.println("File not found!");
        } catch (IOException e) {
            System.out.println("An error occurred while reading the file.");
        }
    }
}
```

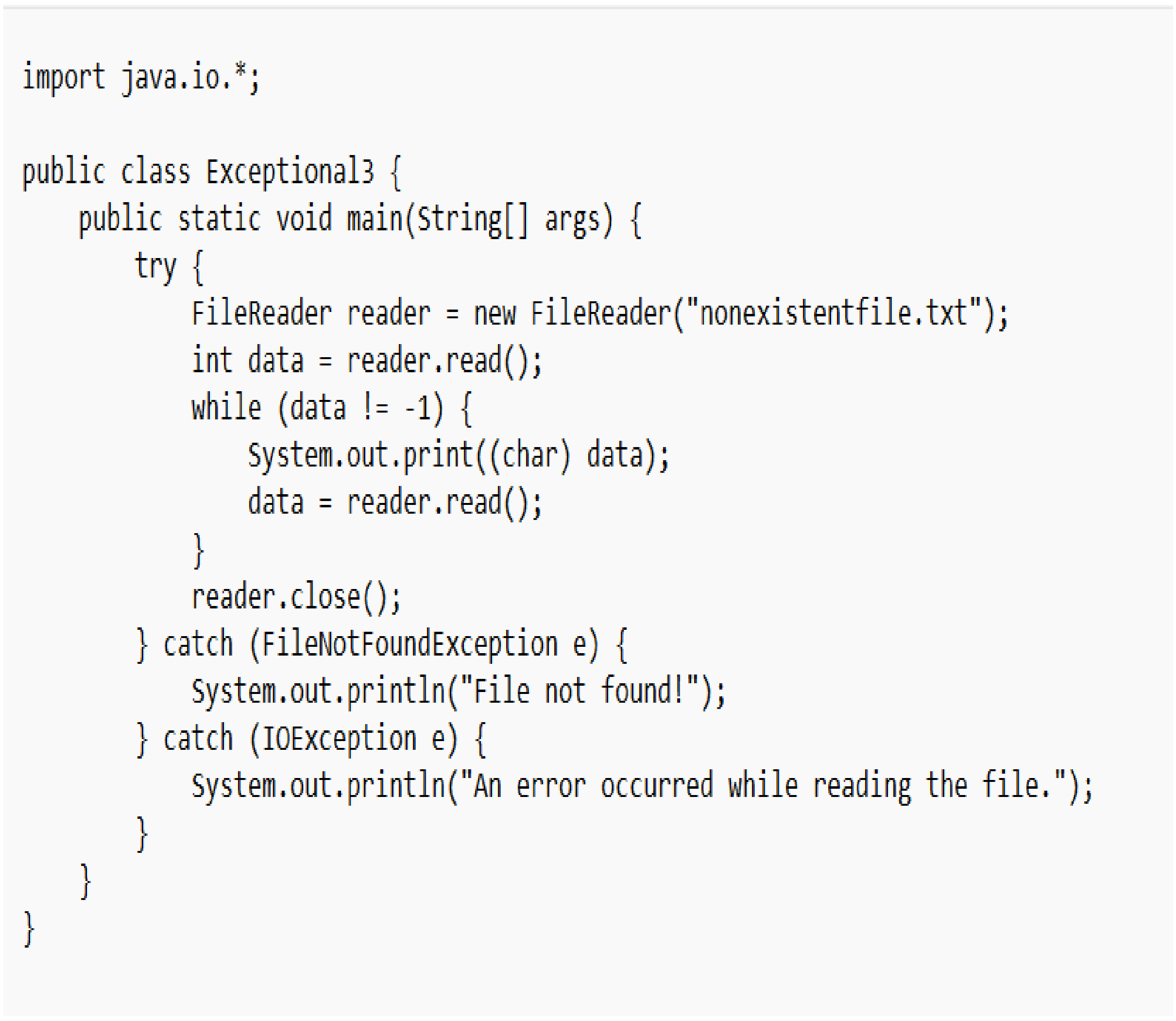
OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac Exceptional2.java

C:\Users\PREETHI JASMINE\Desktop>java Exceptional2
File not found!
```

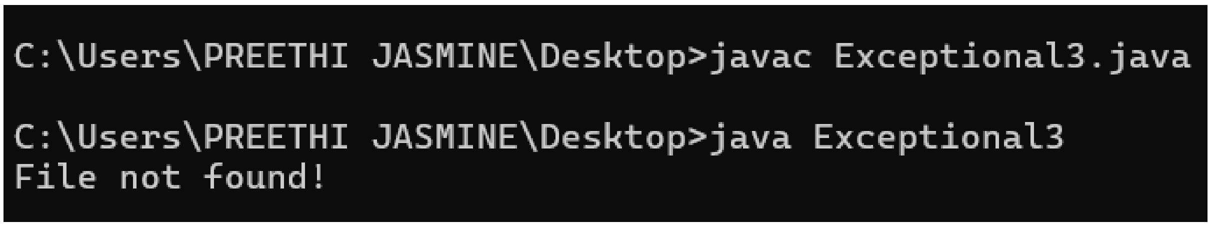
C) File Example2

CODE:



```
import java.io.*;

public class Exceptional3 {
    public static void main(String[] args) {
        try {
            FileReader reader = new FileReader("nonexistentfile.txt");
            int data = reader.read();
            while (data != -1) {
                System.out.print((char) data);
                data = reader.read();
            }
            reader.close();
        } catch (FileNotFoundException e) {
            System.out.println("File not found!");
        } catch (IOException e) {
            System.out.println("An error occurred while reading the file.");
        }
    }
}
```

OUTPUT:



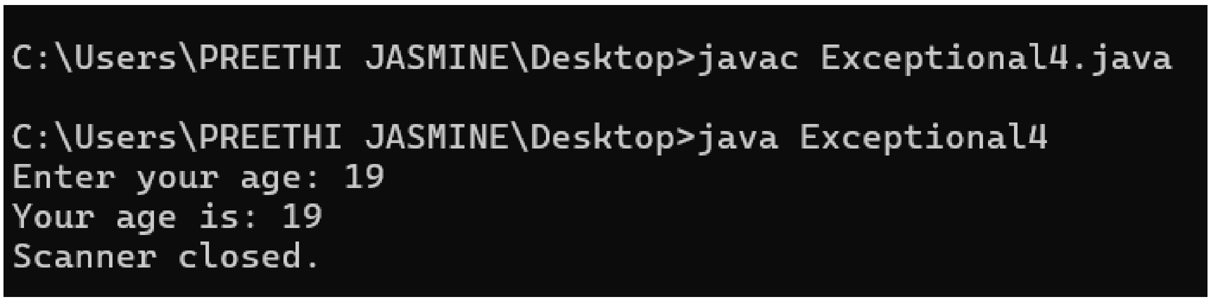
```
C:\Users\PREETHI JASMINE\Desktop>javac Exceptional3.java

C:\Users\PREETHI JASMINE\Desktop>java Exceptional3
File not found!
```

D) Input Validation

CODE:

```java
import java.util.InputMismatchException;
import java.util.Scanner;

public class Exceptional4 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int age = 0;

        try {
            System.out.print("Enter your age: ");
            age = scanner.nextInt();

            if (age < 0) {
                throw new IllegalArgumentException("Age cannot be negative.");
            }

            System.out.println("Your age is: " + age);

        } catch (InputMismatchException e) {
            System.out.println("Invalid input! Please enter a number.");
        } catch (IllegalArgumentException e) {
            System.out.println("Validation error: " + e.getMessage());
        } finally {
            scanner.close();
            System.out.println("Scanner closed.");
        }
    }
}
```

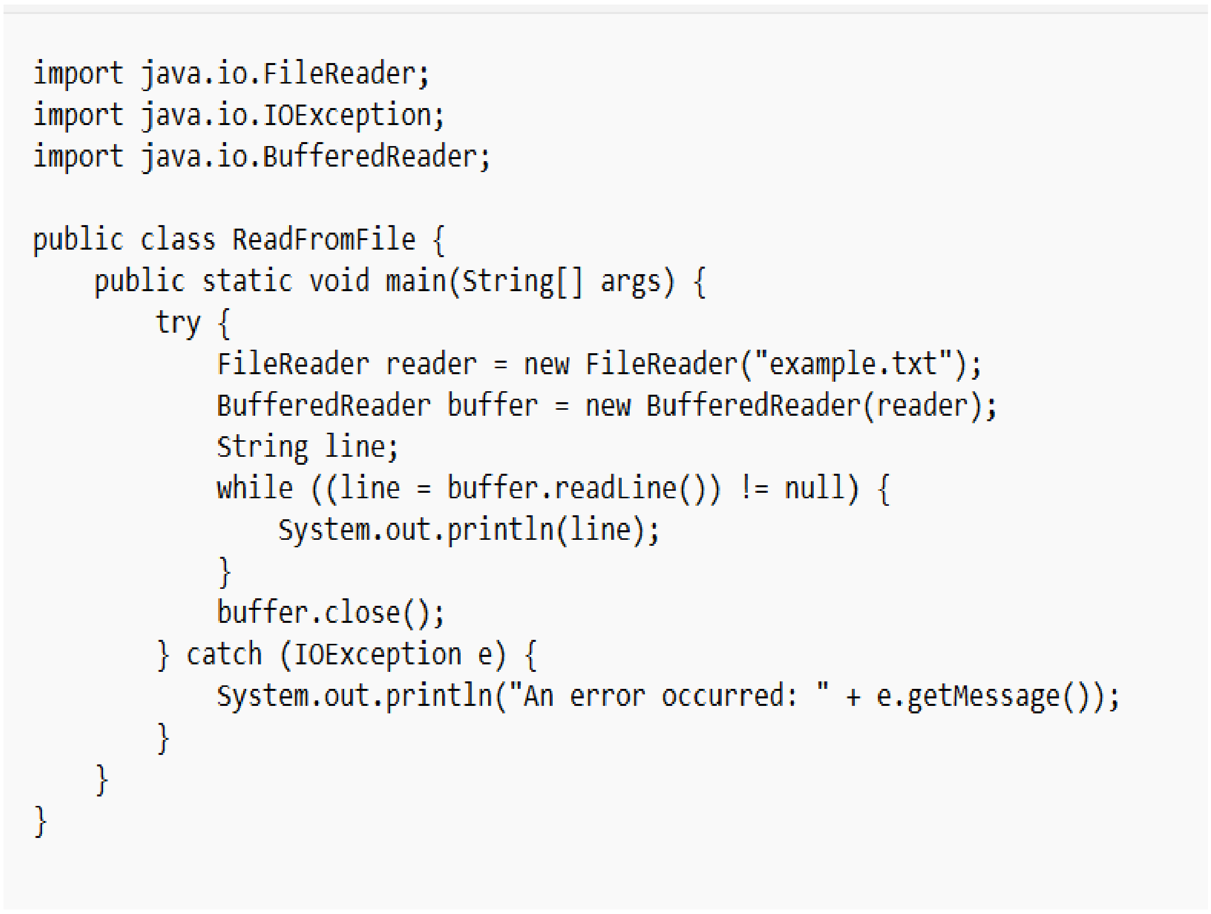
OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac Exceptional4.java

C:\Users\PREETHI JASMINE\Desktop>java Exceptional4
Enter your age: 19
Your age is: 19
Scanner closed.
```

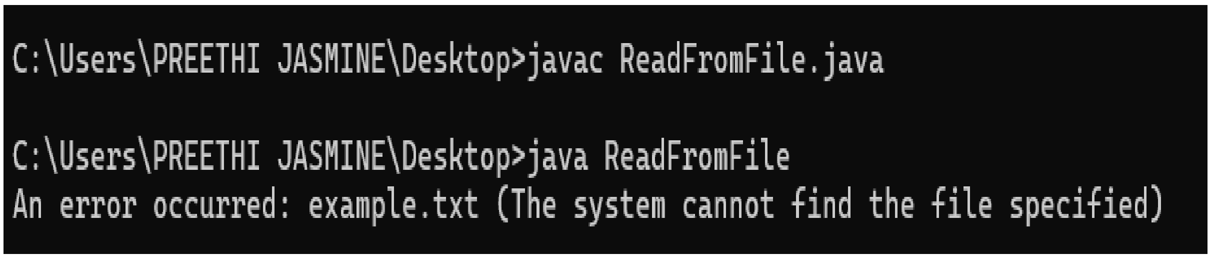
17) FILE HANDLING PROGRAMS

A) Read from File
CODE:



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

public class ReadFromFile {
    public static void main(String[] args) {
        try {
            FileReader reader = new FileReader("example.txt");
            BufferedReader buffer = new BufferedReader(reader);
            String line;
            while ((line = buffer.readLine()) != null) {
                System.out.println(line);
            }
            buffer.close();
        } catch (IOException e) {
            System.out.println("An error occurred: " + e.getMessage());
        }
    }
}
```

OUTPUT:



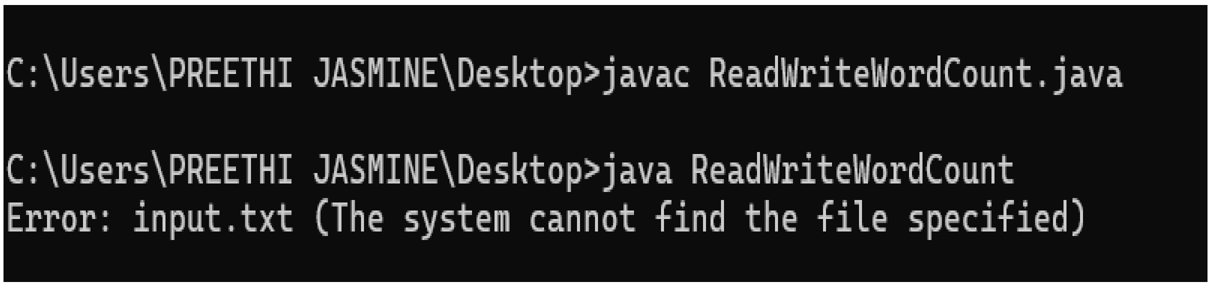
```
C:\Users\PREETHI JASMINE\Desktop>javac ReadFromFile.java

C:\Users\PREETHI JASMINE\Desktop>java ReadFromFile
An error occurred: example.txt (The system cannot find the file specified)
```

B) Word Count

CODE:

```java
import java.io.*;

public class ReadWriteWordCount {
    public static void main(String[] args) {
        try (FileReader file = new FileReader("input.txt");
             BufferedReader reader = new BufferedReader(file);
             FileWriter fileWriter = new FileWriter("output.txt");
             BufferedWriter writer = new BufferedWriter(fileWriter)) {

            String line;
            int wordCount = 0;

            while ((line = reader.readLine()) != null) {
                System.out.println(line);
                writer.write(line);
                writer.newLine();
                wordCount += line.split("\\s+").length; // Count words
            }

            System.out.println("Total Words: " + wordCount);

        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

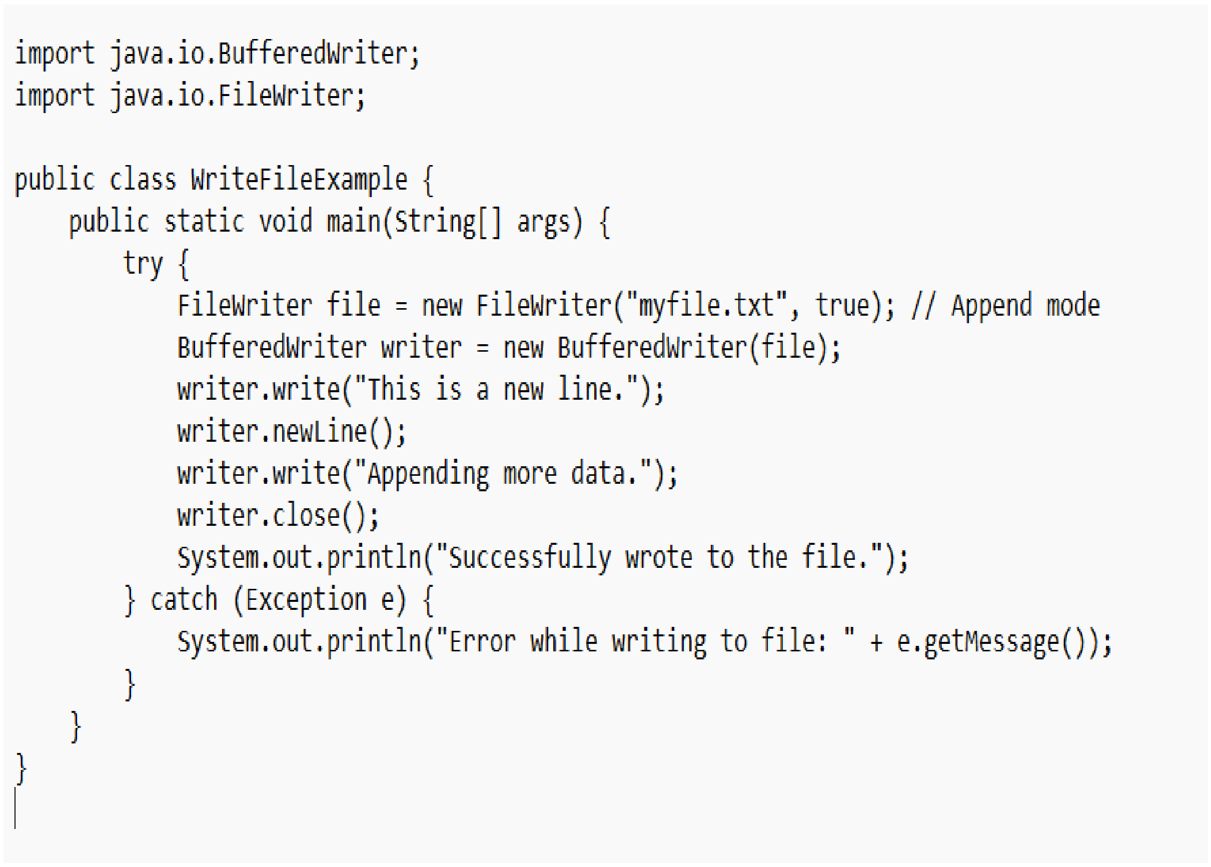
OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac ReadWriteWordCount.java

C:\Users\PREETHI JASMINE\Desktop>java ReadWriteWordCount
Error: input.txt (The system cannot find the file specified)
```

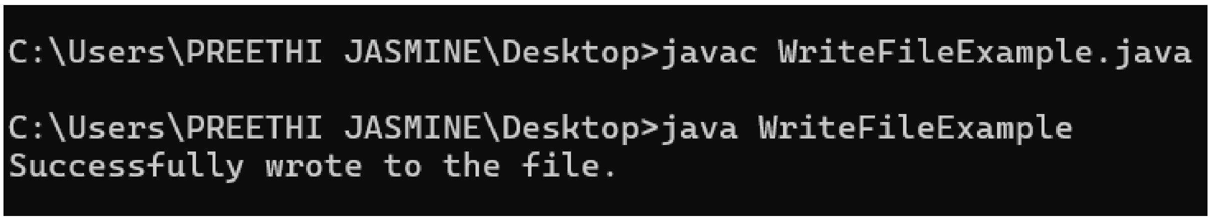
C)File Writer

CODE:



```java
import java.io.BufferedWriter;
import java.io.FileWriter;

public class WriteFileExample {
    public static void main(String[] args) {
        try {
            FileWriter file = new FileWriter("myfile.txt", true); // Append mode
            BufferedWriter writer = new BufferedWriter(file);
            writer.write("This is a new line.");
            writer.newLine();
            writer.write("Appending more data.");
            writer.close();
            System.out.println("Successfully wrote to the file.");
        } catch (Exception e) {
            System.out.println("Error while writing to file: " + e.getMessage());
        }
    }
}
```

OUTPUT:



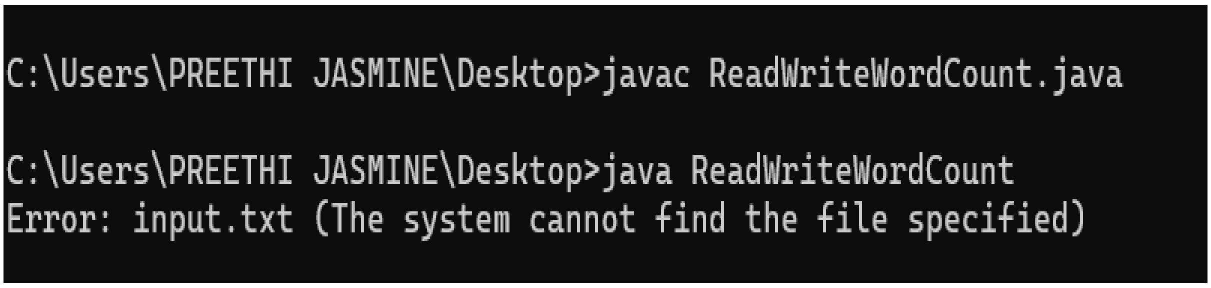
```
C:\Users\PREETHI JASMINE\Desktop>javac WriteFileExample.java

C:\Users\PREETHI JASMINE\Desktop>java WriteFileExample
Successfully wrote to the file.
```

D)File Example 2

CODE:

```java
import java.io.*;

public class ReadWriteWordCount {
    public static void main(String[] args) {
        try (FileReader file = new FileReader("input.txt");
             BufferedReader reader = new BufferedReader(file);
             FileWriter fileWriter = new FileWriter("output.txt");
             BufferedWriter writer = new BufferedWriter(fileWriter)) {

            String line;
            int wordCount = 0;

            while ((line = reader.readLine()) != null) {
                System.out.println(line);
                writer.write(line);
                writer.newLine();
                wordCount += line.split("\\s+").length; // Count words
            }

            System.out.println("Total Words: " + wordCount);

        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

OUTPUT:



```
C:\Users\PREETHI JASMINE\Desktop>javac ReadWriteWordCount.java

C:\Users\PREETHI JASMINE\Desktop>java ReadWriteWordCount
Error: input.txt (The system cannot find the file specified)
```