LAB RECORD

23CSE111 – Object Oriented Programming

Submitted by

CH.SC.U4CSE24107– CHEEDELLA JYOTHIRMAI

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING

CHENNAI

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.U4CSE24107 – CHEEDELLA JYOTHIRMAI*** 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.

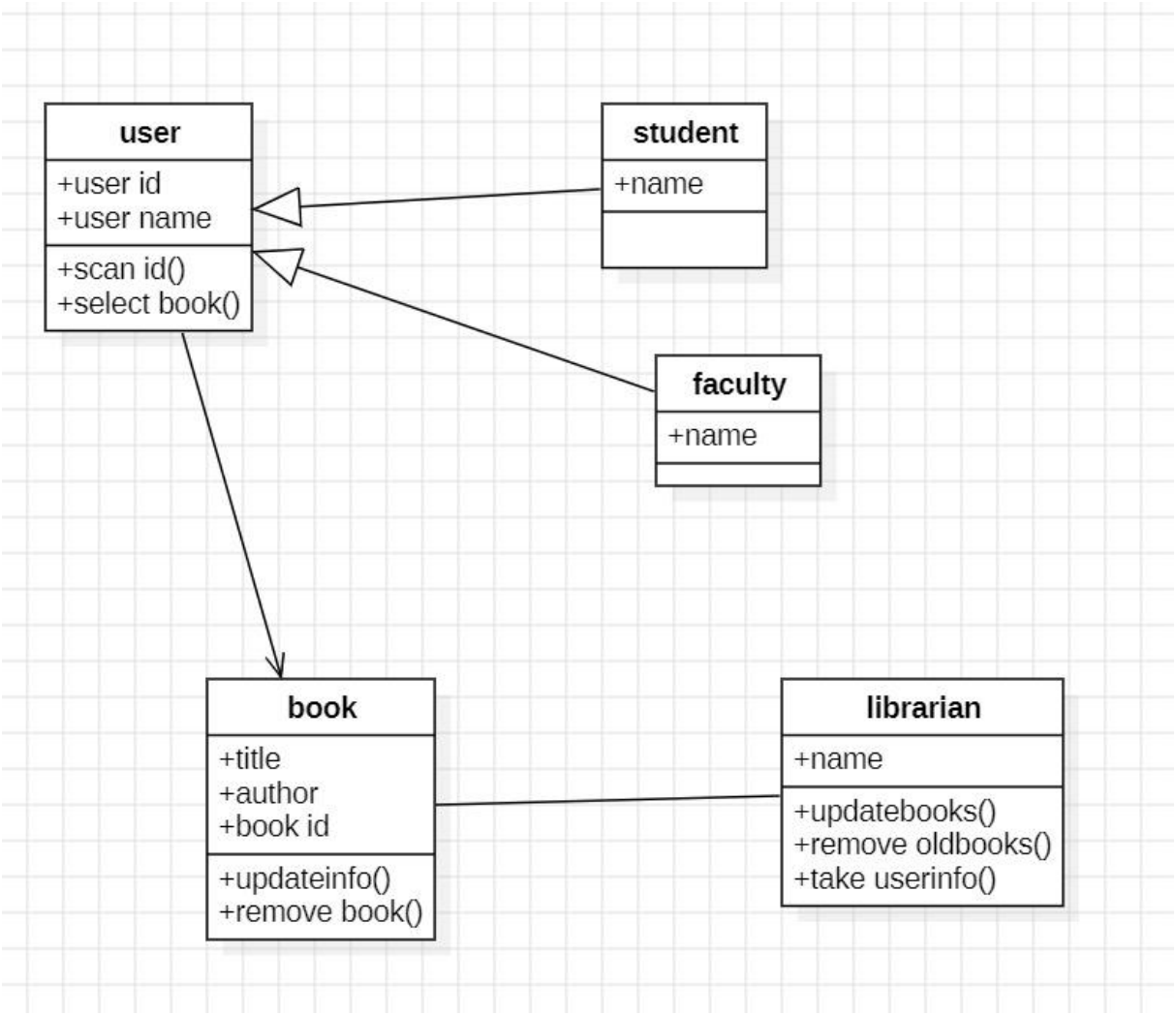
Internal Examiner 1 Internal Examiner 2

Index

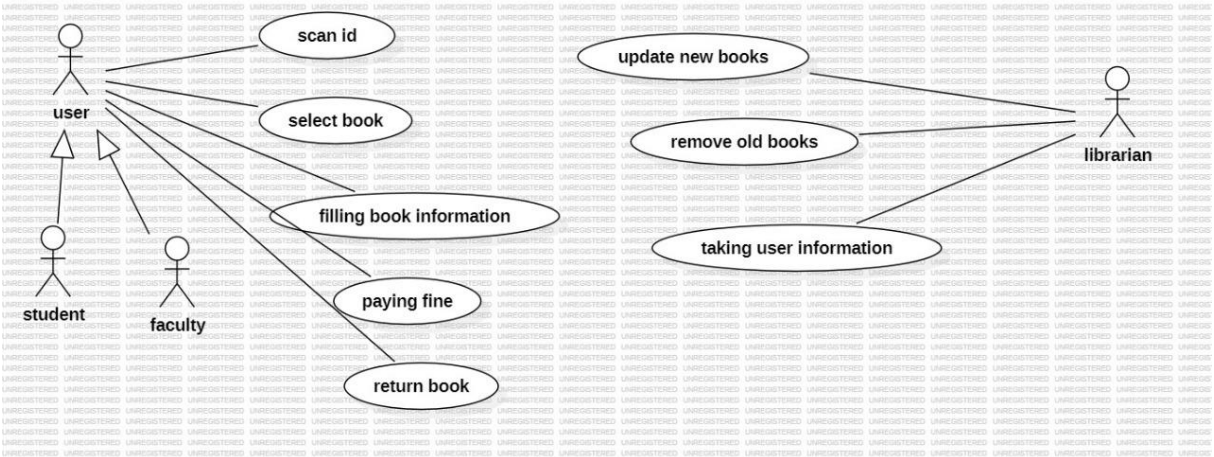
UML DIAGRAM

1.Library Management:

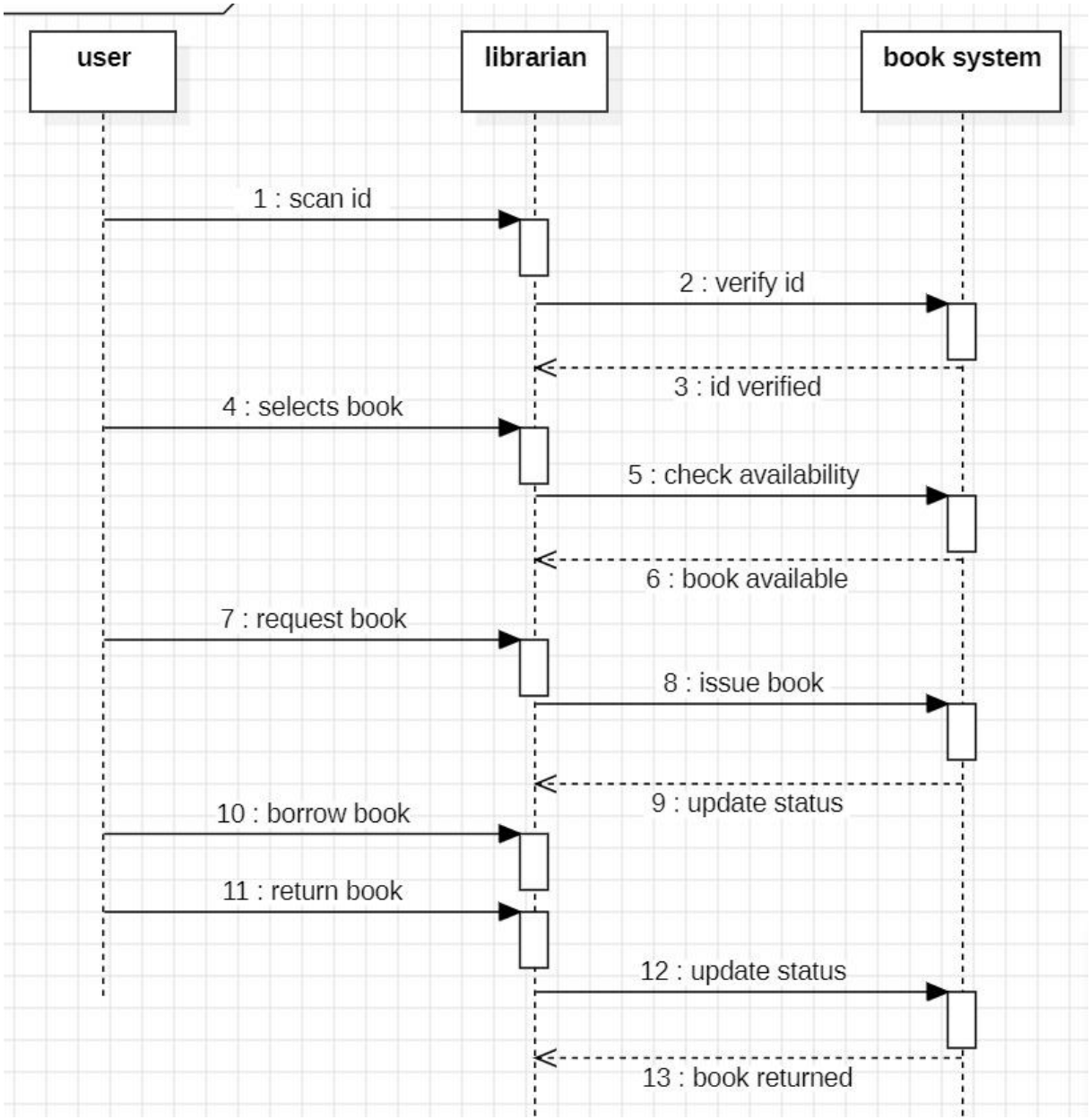
i. Class Diagram:



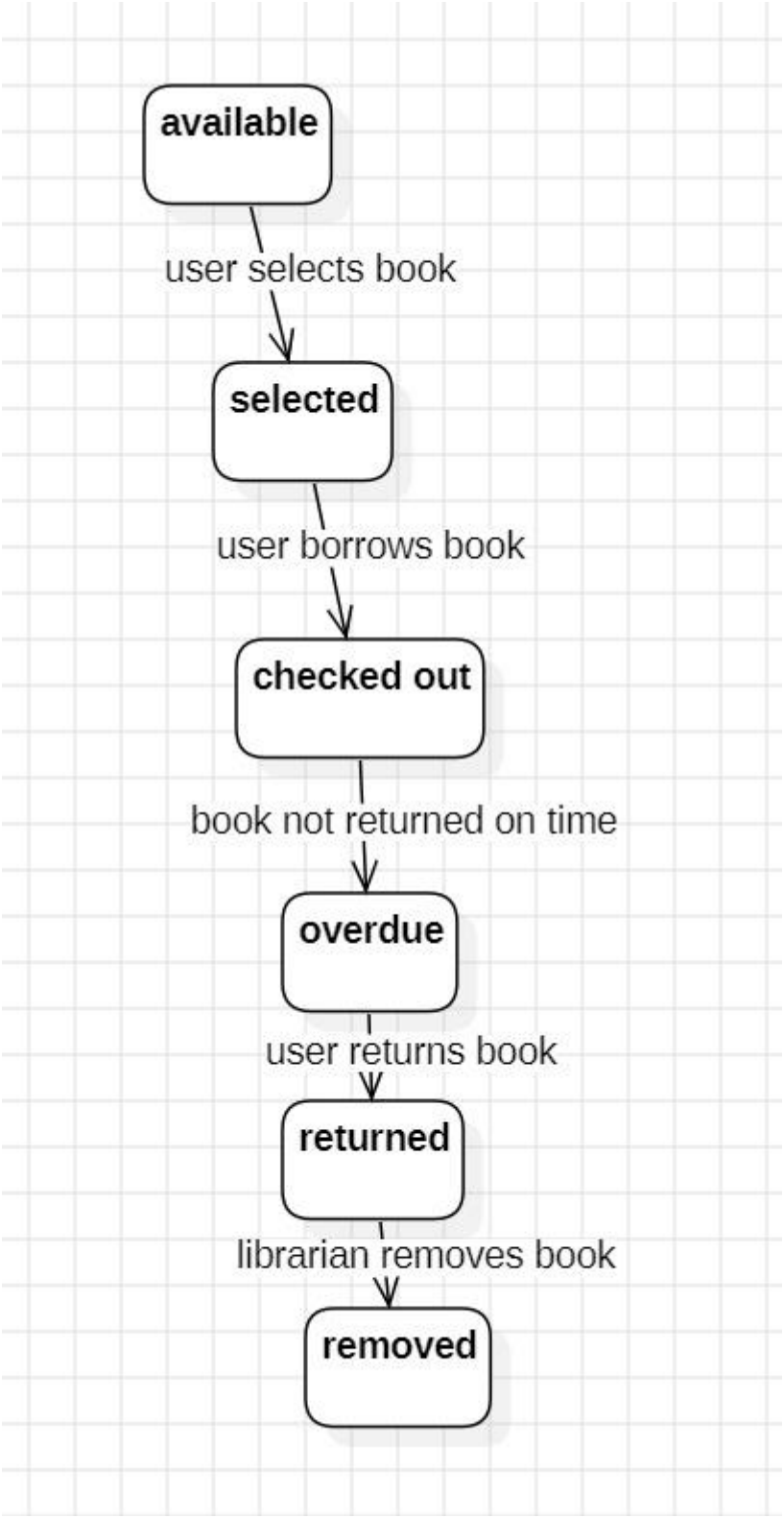
ii. Use Case Diagram:



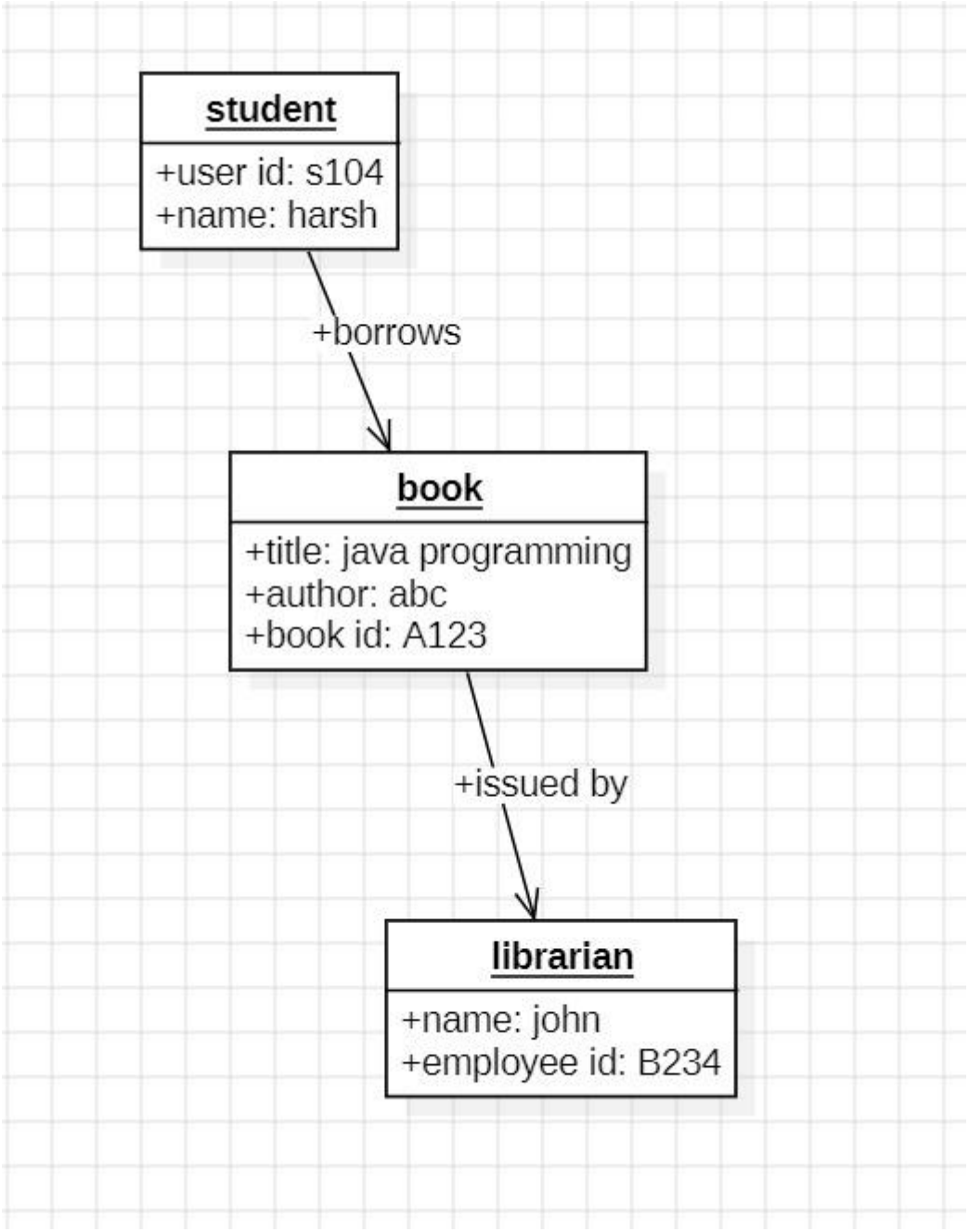
iii. Sequence Diagram:


user
librarian
book system
1 : scan id
2 : verify id
3 : id verified
4 : selects book
5 : check availability
6 : book available
7 : request book
8 : issue book
9 : update status
10 : borrow book
11 : return book
12 : update status
13 : book returned

iv. State Activity Diagram:

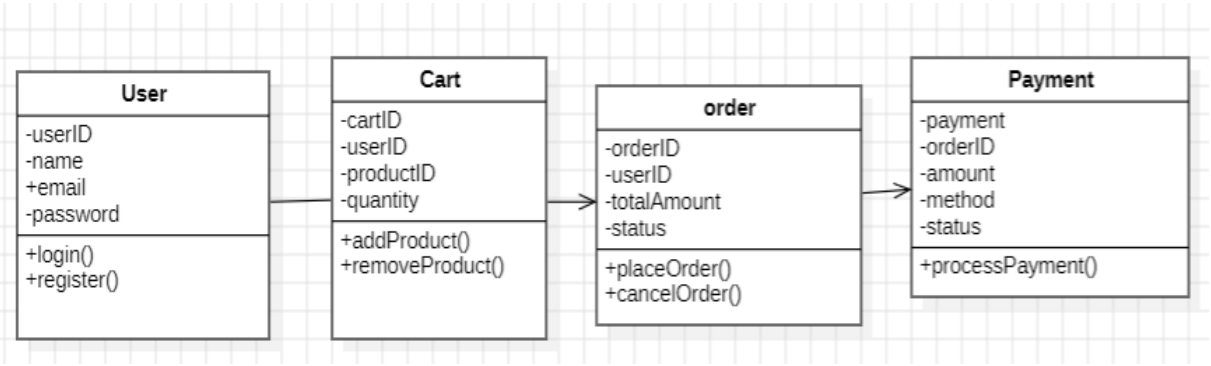


v. Object Diagram:

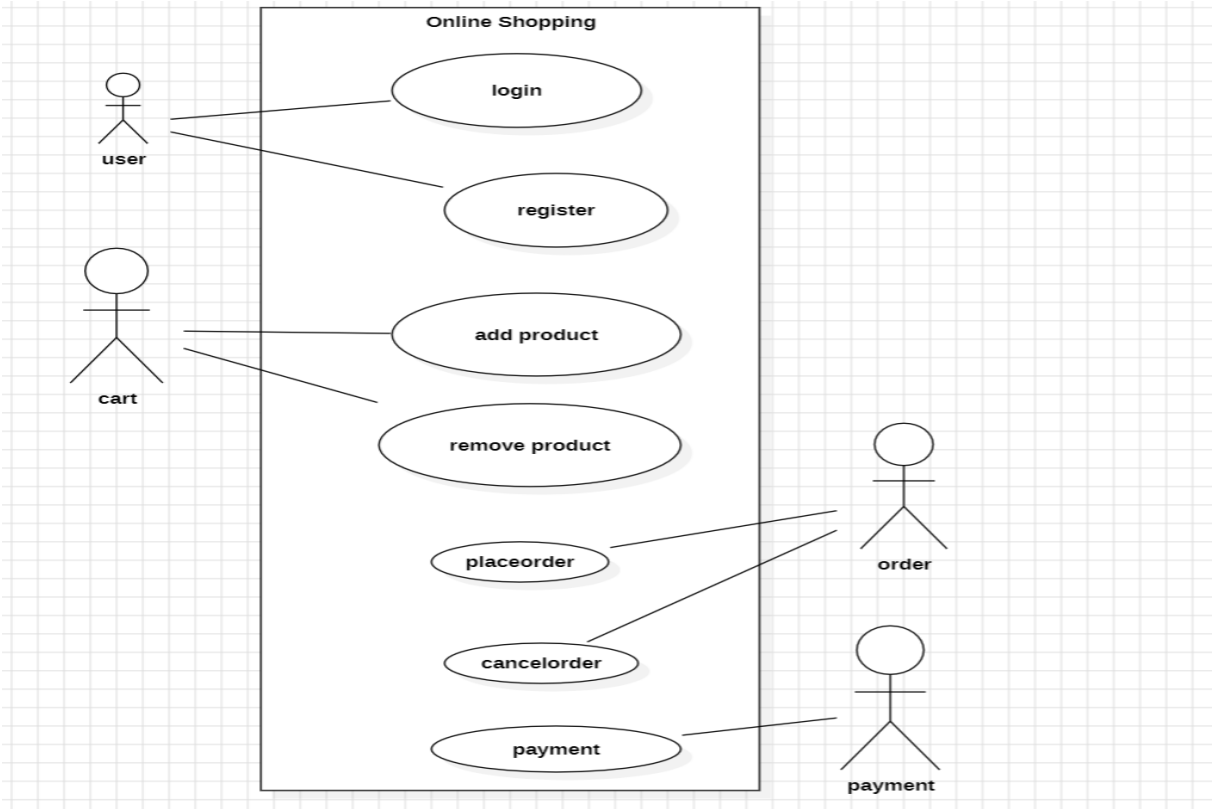


2.Online Shopping

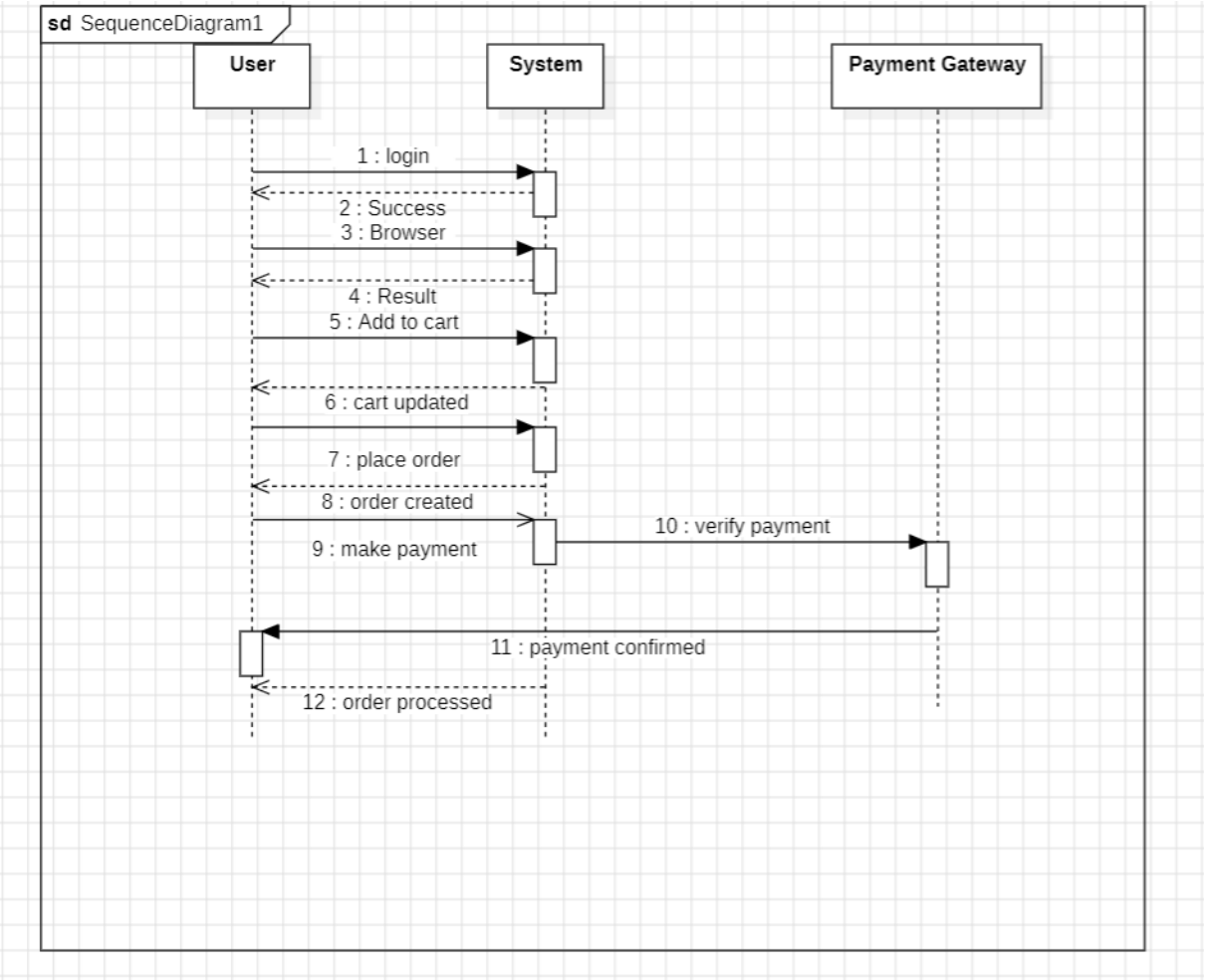
i. Class Diagram:



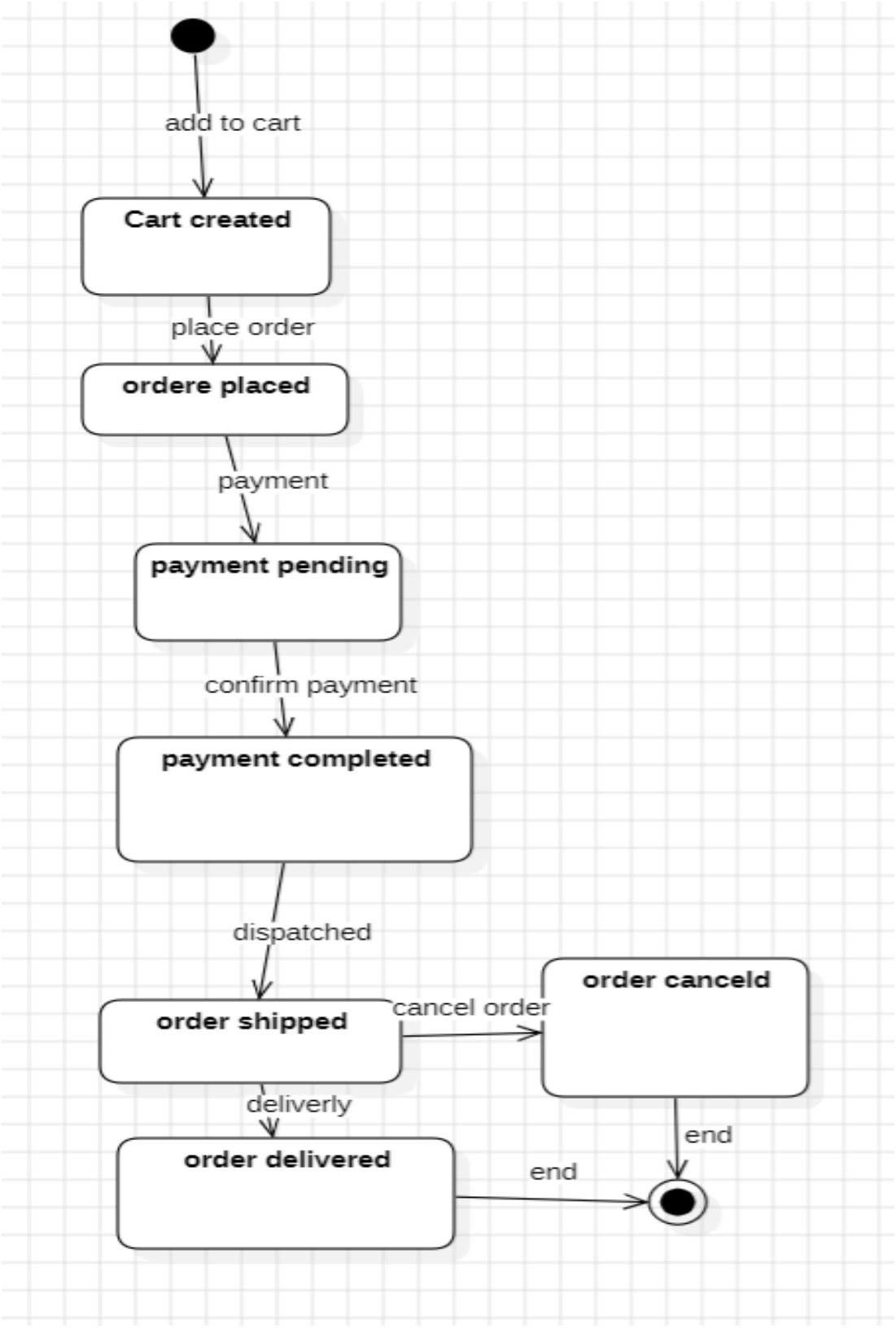
ii. Use Case Diagram:



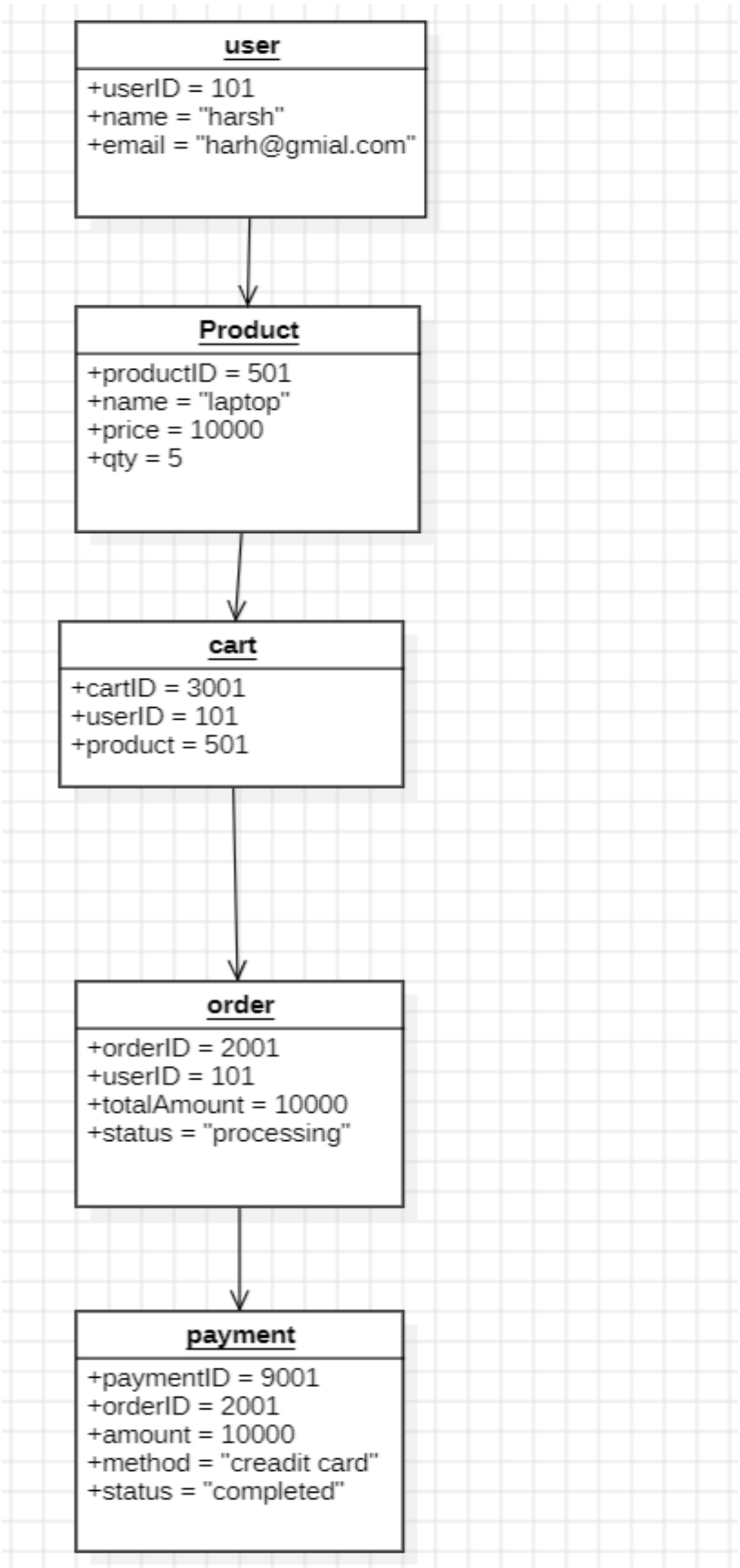
iii. Sequence Diagram:



iv. State Activity Diagram:



v. Object Diagram:


user
+userID = 101
+name = "harsh"
+email = "harh@gmial.com"
Product
+productID = 501
+name = "laptop"
+price = 10000
+qty = 5
cart
+cartID = 3001
+userID = 101
+product = 501
order
+orderID = 2001
+userID = 101
+totalAmount = 10000
+status = "processing"
payment
+paymentID = 9001
+orderID = 2001
+amount = 10000
+method = "creadit card"
+status = "completed"

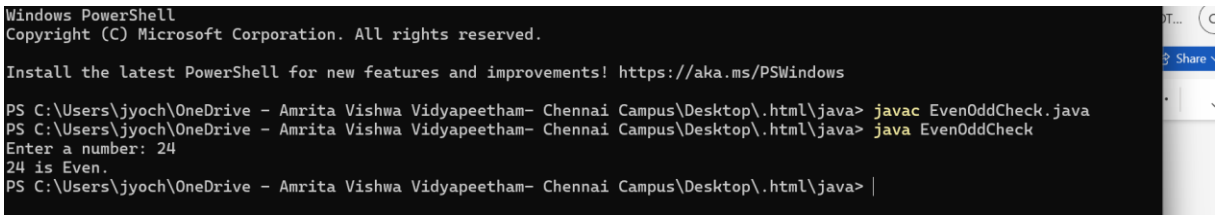
Basic Java Programs

1. 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:

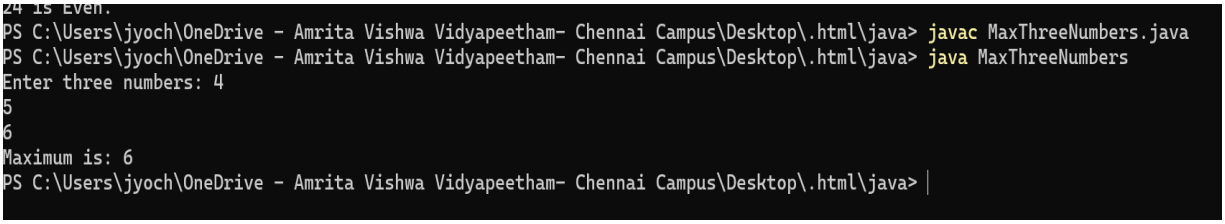


2. Max of 3 numbers:

Code:

```java
import java.util.Scanner;

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

        if (a > b && a > c)
            System.out.println("Maximum is: " + a);
        else if (b > c)
            System.out.println("Maximum is: " + b);
        else
            System.out.println("Maximum is: " + c);

        sc.close();
    }
}
```

Output:



3.Calculator

Code:

```java
import java.util.Scanner;

public class Calculator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter first number: ");
        double num1 = sc.nextDouble();
        System.out.print("Enter second number: ");
        double num2 = sc.nextDouble();
        System.out.print("Enter operation (+, -, *, /): ");
        char op = sc.next().charAt(0);

        double result;
        switch (op) {
            case '+': result = num1 + num2; break;
            case '-': result = num1 - num2; break;
            case '*': result = num1 * num2; break;
            case '/':
                if (num2 != 0) result = num1 / num2;
                else { System.out.println("Cannot divide by zero!"); return; }
                break;
            default: System.out.println("Invalid operator!"); return;
        }

        System.out.println("Result: " + result);
        sc.close();
    }
}
```

Output:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac Calculator.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java Calculator
Enter first number: 23
Enter second number: 65
Enter operation (+, -, *, /): *
Result: 1495.0
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java>
```

4.Fibonacci series:

Code:

```
import java.util.Scanner;

public class FibonacciSeries {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter 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:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac FibonacciSeries.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java  FibonacciSeries
Enter number of terms: 4
Fibonacci Series: 0 1 1 2
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java>
```

5.Reverse a number:

Code:

```java
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 rev = 0;

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

        System.out.println("Reversed number: " + rev);
        sc.close();
    }
}
```

Output:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac ReverseNumber
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java ReverseNumber
Enter a number: 536
Reversed number: 635
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java>
```

6.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:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac Factorial.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java Factorial
Enter a number: 6
Factorial of 6 is: 720
```

7.Prime Number Check:

Code:

```java
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 <= num / 2; i++) {
                if (num % i == 0) {
                    isPrime = 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:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac PrimeCheck.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java PrimeCheck
Enter a number: 12
12 is not a Prime number.
```

8.Palindrome Check:

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, rev = 0;

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

        if (original == rev)
            System.out.println(original + " is a Palindrome.");
        else
            System.out.println(original + " is not a Palindrome.");

        sc.close();
    }
}
```

Output:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac PalindromeCheck.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java PalindromeCheck
Enter a number: 23
23 is not a Palindrome.
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java>
```

9.Armstrong Number:

Code:

```java
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 = String.valueOf(num).length();

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

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac  ArmstrongNumber.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java  ArmstrongNumber
Enter a number: 153
153 is an Armstrong number.
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java>
```

10.Sum of Natural Number:

Code:

```java
import java.util.Scanner;

public class SumNaturalNumbers {
    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;
        for (int i = 1; i <= num; i++) {
            sum += i;
        }

        System.out.println("Sum of first " + num + " natural numbers is: " + sum);
        sc.close();
    }
}
```

Output:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac SumNaturalNumbers.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> ja
                                                                                            java SumNaturalNumbers
Enter a number: 5
```

INHERITANCE:

4)Hierarchical Inheritance:

A)

CODE:

```
class Vehicle {
   private String brand;
   private String model;
    public Vehicle(String brand, String model) {
      this.brand = brand;
      this.model = model;
   }
   public void start() {
      System.out.println("Vehicle is starting.");
   }
   public void stop() {
      System.out.println("Vehicle is stopping.");
```

```
    }
    public String getBrand() {
        return brand;
    }
  public String getModel() {
        return model;
    }
}
class Car extends Vehicle {
    private int numberOfDoors
   public Car(String brand, String model, int numberOfDoors) {
        super(brand, model);
        this.numberOfDoors = numberOfDoors;
    }
  public void drive() {
        System.out.println("Car is driving.");
    }
    public int getNumberOfDoors() {
        return numberOfDoors;
    }
}
class ElectricCar extends Car {
    private int batteryCapacity;
   public ElectricCar(String brand, String model, int numberOfDoors, int batteryCapacity)
{
        super(brand, model, numberOfDoors);
        this.batteryCapacity = batteryCapacity;
    }
  public void charge() {
        System.out.println("Electric car is charging.");
```

```
    }
 public int getBatteryCapacity() {
        return batteryCapacity;
    }
}
class Truck extends Vehicle {
    private double cargoCapacity;
    public Truck(String brand, String model, double cargoCapacity) {
        super(brand, model);
        this.cargoCapacity = cargoCapacity;
    }
 public void loadCargo() {
        System.out.println("Truck is loading cargo.");
    }
  public double getCargoCapacity() {
        return cargoCapacity;
    }
}
public class Main {
    public static void main(String[] args) {
        Car car = new Car("Toyota", "Corolla", 4);
        car.start();
        car.drive();
        car.stop();
        System.out.println("Car doors: " + car.getNumberOfDoors());
        ElectricCar electricCar = new ElectricCar("Tesla", "Model S", 4, 100);
        electricCar.start();
        electricCar.drive();
        electricCar.charge();
        System.out.println("Battery capacity: " + electricCar.getBatteryCapacity());
```

```
    Truck truck = new Truck("Ford", "F-150", 2000.5);
    truck.start();
    truck.loadCargo();
    truck.stop();
    System.out.println("Cargo capacity: " + truck.getCargoCapacity());
  }
}
```

OUTPUT:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop> javac q1.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop> java q1
Vehicle is starting.
Car is driving.
Vehicle is stopping.
Car doors: 4
Vehicle is starting.
Car is driving.
Electric car is charging.
Battery capacity: 100
Vehicle is starting.
Truck is loading cargo.
Vehicle is stopping.
Cargo capacity: 2000.5
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop>
```

B)

CODE:

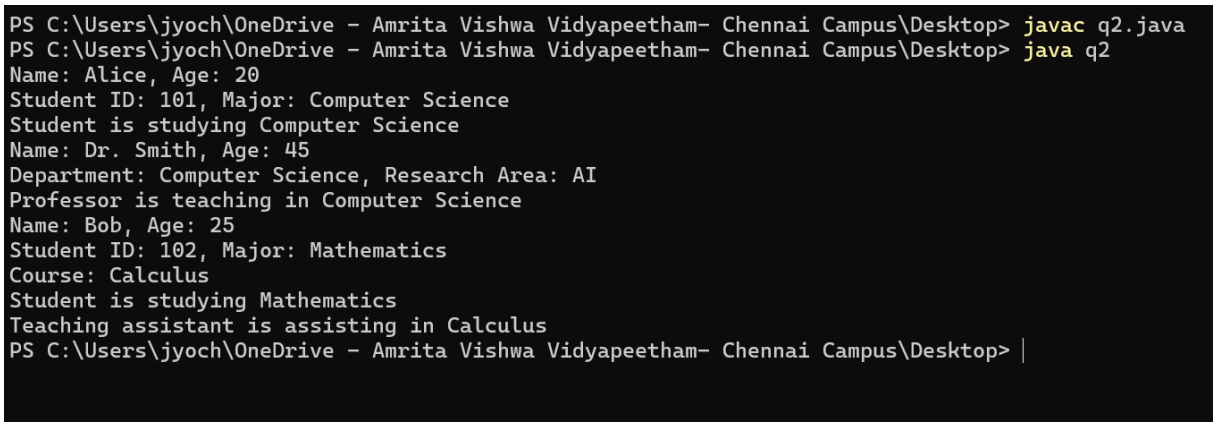
```
class Person {
  private String name;
  private int age;

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

  public void displayDetails() {
    System.out.println("Name: " + name + ", Age: " + age);
```

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

```
        super.displayDetails();
        System.out.println("Department: " + department + ", Research Area: " +
researchArea);
    }
}
class TeachingAssistant extends Student {
    private String course;

    public TeachingAssistant(String name, int age, int studentId, String major, String
course) {
        super(name, age, studentId, major);
        this.course = course;
    }
    public void assist() {
        System.out.println("Teaching assistant is assisting in " + course);
    }
public void displayDetails() {
        super.displayDetails();
        System.out.println("Course: " + course);
    }
}
public class q2 {
    public static void main(String[] args) {
        Student student = new Student("Alice", 20, 101, "Computer Science");
        student.displayDetails();
        student.study()
         Professor professor = new Professor("Dr. Smith", 45, "Computer Science", "AI");
        professor.displayDetails();
        professor.teach(TeachingAssistant ta = new TeachingAssistant("Bob", 25, 102,
"Mathematics", "Calculus");
        ta.displayDetails();
```

```
    ta.study();
    ta.assist();
  }
}
```

OUTPUT:



5)SINGLE INHERITANCE:

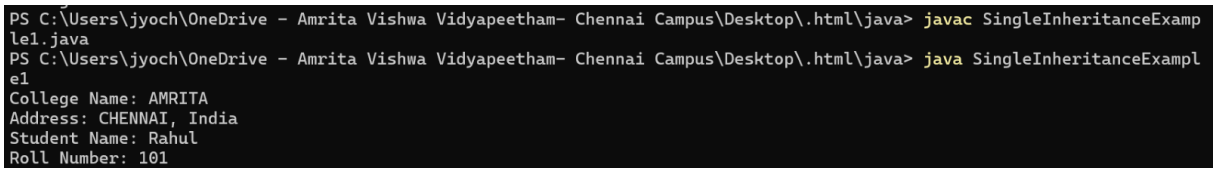
A)

CODE:

```
class College {
  String collegeName = "AMRITA";
  String address = "CHENNAI, India";

  void showCollegeDetails() {
    System.out.println("College Name: " + collegeName);
    System.out.println("Address: " + address);
  }
}
class Student extends College {
  String studentName;
  int rollNumber;
```

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

    void showStudentDetails() {
        System.out.println("Student Name: " + studentName);
        System.out.println("Roll Number: " + rollNumber);
    }
}
public class SingleInheritanceExample1 {
    public static void main(String[] args) {
        Student s1 = new Student("Rahul", 101);
        s1.showCollegeDetails();
        s1.showStudentDetails();
    }
}
```

OUTPUT:



B)

CODE:

```
class BankAccount {
    String accountHolder;
    double balance;
```

```
    BankAccount(String accountHolder, double balance) {
        this.accountHolder = accountHolder;
        this.balance = balance;
    }

    void showBalance() {
        System.out.println("Account Holder: " + accountHolder);
        System.out.println("Balance: $" + balance);
    }
}
class SavingsAccount extends BankAccount {
    double interestRate = 5.0;

    SavingsAccount(String accountHolder, double balance) {
        super(accountHolder, balance);
    }

    void calculateInterest() {
        double interest = (balance * interestRate) / 100;
        System.out.println("Annual Interest: $" + interest);
    }
}

public class SingleInheritanceExample2 {
    public static void main(String[] args) {
        SavingsAccount acc1 = new SavingsAccount("John Doe", 5000);
        acc1.showBalance();
        acc1.calculateInterest();
    }
}
```

OUTPUT:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac SingleInheritanceExamp
le2.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java SingleInheritanceExampl
e2
Account Holder: John Doe
Balance: $5000.0
Annual Interest: $250.0
```

6)MULTILEVEL INHERITANCE:

A)

CODE:

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

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

   void study() {
```

```
    System.out.println(name + " is studying.");
  }

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

OUTPUT:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac  MultilevelExample1.ja
va
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java  MultilevelExample1
Living beings breathe.
Humans can speak.
Rahul is studying.
Student Name: Rahul
Student ID: 101
```

B)

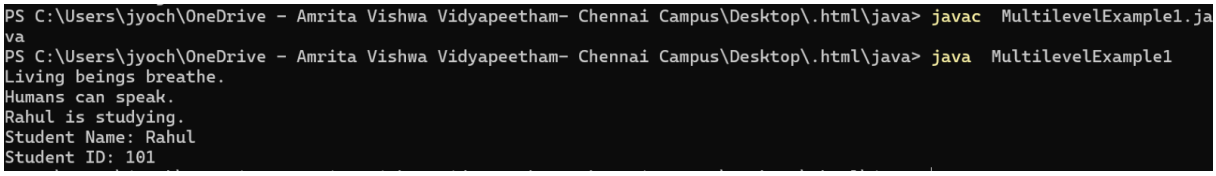
CODE:

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

```
    void showPerson() {
        System.out.println("Person Name: " + name);
    }
}
class Employee extends Person {
    int employeeID;
    double salary;
    Employee(String name, int employeeID, double salary) {
        super(name);
        this.employeeID = employeeID;
        this.salary = salary;
    }
    void showEmployee() {
        System.out.println("Employee ID: " + employeeID);
        System.out.println("Salary: $" + salary);
    }
}
class Manager extends Employee {
    String department;
    Manager(String name, int employeeID, double salary, String department) {
        super(name, employeeID, salary);
        this.department = department;
    }
    void showManager() {
        System.out.println("Department: " + department);
        System.out.println("Role: Manager");
    }
}
public class MultilevelExample2 {
    public static void main(String[] args) {
```

```
    Manager m1 = new Manager("Alice", 2001, 75000, "HR");
    m1.showPerson();
    m1.showEmployee();
    m1.showManager();
  }
}
```

OUTPUT:



7)HYBRID INHERITANCE:

A)

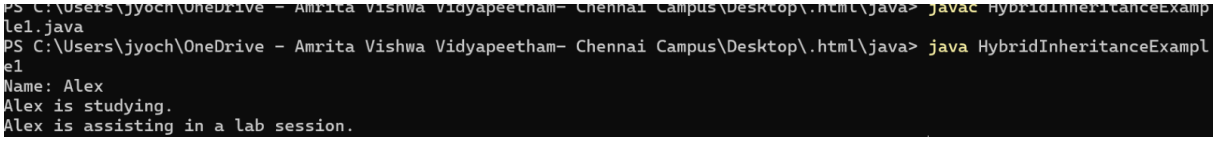
CODE:

```
class Person {
  String name;

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

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

  Student(String name, int studentID) {
```

```
        super(name);
        this.studentID = studentID;
    }

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

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

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

    public void assist() {
        System.out.println(name + " is assisting in a lab session.");
```

```
    }
}
public class HybridInheritanceExample1 {
    public static void main(String[] args) {
        TeachingAssistant ta = new TeachingAssistant("Alex", 101);
        ta.showDetails();
        ta.study();
        ta.assist();
    }
}
```

OUTPUT:



B)

CODE:

```
class Vehicle {
    void startEngine() {
        System.out.println("Vehicle engine started.");
    }
}
class Car extends Vehicle {
    void drive() {
        System.out.println("Car is driving.");
    }
}
class Boat extends Vehicle {
    void sail() {
```

```
        System.out.println("Boat is sailing.");
    }
}
interface Amphibious {
    void switchMode();
}
class AmphibiousCar extends Car implements Amphibious {
    public void switchMode() {
        System.out.println("Switching between land and water mode.");
    }

    void sail() {
        System.out.println("Amphibious car is sailing on water.");
    }
}
public class HybridInheritanceExample2 {
    public static void main(String[] args) {
        AmphibiousCar ac = new AmphibiousCar();
        ac.startEngine();
        ac.drive();
        ac.switchMode();
        ac.sail();
    }
}
```

OUTPUT:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac HybridInheritanceExampl
e2.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java HybridInheritanceExampl
e2
Vehicle engine started.
Car is driving.
Switching between land and water mode.
Amphibious car is sailing on water.
```

POLYMORPHISM:

8)METHOD OVERLOADING

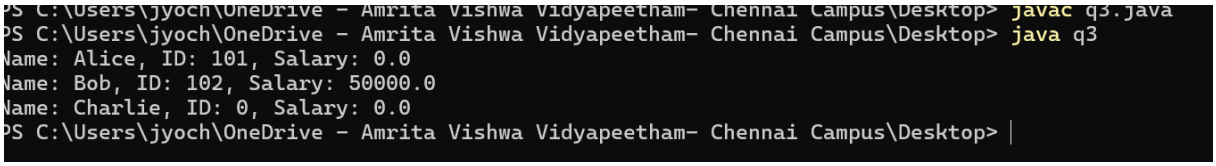
A)

CODE:

```
class Employee {
   private String name;
   private int id;
   private double salary;
   void setDetails(String name, int id) {
      this.name = name;
      this.id = id;
   }
   void setDetails(String name, int id, double salary) {
      this.name = name;
      this.id = id;
      this.salary = salary;
   }
   void setDetails(String name) {
      this.name = name;
   }
   void displayDetails() {
      System.out.println("Name: " + name + ", ID: " + id + ", Salary: " + salary);
   }
}
public class q3 {
   public static void main(String[] args) {
      Employee emp1 = new Employee();
      emp1.setDetails("Alice", 101); // Calls first method
      emp1.displayDetails();
```

```
        Employee emp2 = new Employee();
        emp2.setDetails("Bob", 102, 50000.0); // Calls second method
        emp2.displayDetails();
        Employee emp3 = new Employee();
        emp3.setDetails("Charlie"); // Calls third method
        emp3.displayDetails();
    }
}
```

OUTPUT:



B)

CODE:

```
class Shape {
    void draw(double radius) {
        System.out.println("Drawing a circle with radius: " + radius);
    }
  void draw(double length, double width) {
        System.out.println("Drawing a rectangle with length: " + length + " and width: " + width);
    }
    void draw(double side1, double side2, double side3) {
        System.out.println("Drawing a triangle with sides: " + side1 + ", " + side2 + ", " + side3);
    }
}
public class q4 {
    public static void main(String[] args) {
```

```
        Shape shape = new Shape();
        shape.draw(5.0);
        shape.draw(4.0, 6.0);
        shape.draw(3.0, 4.0, 5.0);    }
}
```

OUTPUT:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop> javac q4.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop> java q4
Drawing a circle with radius: 5.0
Drawing a rectangle with length: 4.0 and width: 6.0
Drawing a triangle with sides: 3.0, 4.0, 5.0
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop>
```

9)METHOD OVER RIDING:

A)

CODE:

```
abstract class Employee {
    protected String name;
    protected double salary;
    public Employee(String name, double salary) {
        this.name = name;
        this.salary = salary;
    }
    abstract double calculateBonus();
    public void displayDetails() {
        System.out.println("Name: " + name + ", Salary: " + salary);
    }
}
class Manager extends Employee {
    private static final double BONUS_PERCENTAGE = 0.20; // 20% bonus

    public Manager(String name, double salary) {
```

```
        super(name, salary);
    }
    double calculateBonus() {
        return salary * BONUS_PERCENTAGE;
    }
}
class Developer extends Employee {
    private static final double BONUS_PERCENTAGE = 0.10; // 10% bonus
     public Developer(String name, double salary) {
        super(name, salary);
    }
    double calculateBonus() {
        return salary * BONUS_PERCENTAGE;
    }
}
public class q5 {
    public static void main(String[] args) {
        Employee manager = new Manager("John", 100000);
        Employee developer = new Developer("Alice", 80000);
       System.out.println("Manager Bonus: " + manager.calculateBonus());
        System.out.println("Developer Bonus: " + developer.calculateBonus());
      manager.displayDetails();
        developer.displayDetails();
    }
}
```

OUTPUT:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop> javac q5.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop> java q5
Manager Bonus: 20000.0
Developer Bonus: 8000.0
Name: John, Salary: 100000.0
Name: Alice, Salary: 80000.0
```

B)

CODE:

abstract class Shape { abstract double calculateArea(); abstract double calculatePerimeter(); } class Circle extends Shape { private double radius;

```
public Circle(double radius) {
    this.radius = radius;
}
double calculateArea() {
    return Math.PI * radius * radius;
}
double calculatePerimeter() {
    return 2 * Math.PI * radius;
}
```

} class Rectangle extends Shape { private double length; private double width;

```
public Rectangle(double length, double width) {
    this.length = length;
    this.width = width;
}
double calculateArea() {
    return length * width;
}

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

}
```

public class q6 { public static void main(String[] args) {

Shape circle = new Circle(5);

Shape rectangle = new Rectangle(4, 6);

```
   System.out.println("Circle Area: " + circle.calculateArea());
    System.out.println("Circle Perimeter: " +
circle.calculatePerimeter());

    System.out.println("Rectangle Area: " +
rectangle.calculateArea());
    System.out.println("Rectangle Perimeter: " +
rectangle.calculatePerimeter());
}
}
```

OUTPUT:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop> javac q6.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop> java q6
Circle Area: 78.53981633974483
Circle Perimeter: 31.41592653589793
Rectangle Area: 24.0
Rectangle Perimeter: 20.0
```

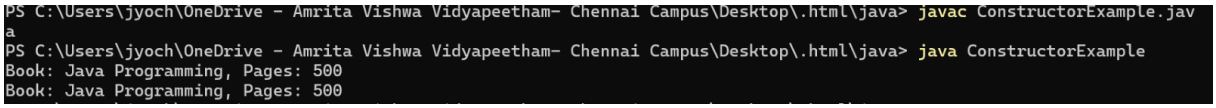
10)CONSTRUCTOR PROGRMS:

CODE:

```
class Book {
   String title;
   int pages;
   Book(String t, int p) {
      title = t;
      pages = p;
   }
   Book(Book b) {
      title = b.title;
      pages = b.pages;
   }
   void display() {
      System.out.println("Book: " + title + ", Pages: " + pages);
   }
}
 public class ConstructorExample {
   public static void main(String[] args) {
      Book b1 = new Book("Java Programming", 500);
      Book b2 = new Book(b1);
      b1.display();
      b2.display();
   }
```

}

OUTPUT:



11)CONSTRUCTOR OVERLOADING:

CODE:

```
class Employee {
   String name;
   int age;
   double salary;

   Employee() {
      name = "Unknown";
      age = 18;
      salary = 30000;
   }

   Employee(String n, int a) {
      name = n;
      age = a;
      salary = 40000;
   }

   Employee(String n, int a, double s) {
      name = n;
      age = a;
      salary = s;
```

```
    }

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

public class ConstructorOverloadingExample {
    public static void main(String[] args) {
        Employee e1 = new Employee();
        Employee e2 = new Employee("John", 25);
        Employee e3 = new Employee("Alice", 30, 60000);

        e1.display();
        e2.display();
        e3.display();
    }
}
```

OUTPUT:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac ConstructorOverloadi
Example.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java ConstructorOverloadin
xample
Name: Unknown, Age: 18, Salary: $30000.0
Name: John, Age: 25, Salary: $40000.0
Name: Alice, Age: 30, Salary: $60000.0
```

ABSTRACTION:

12)ABSTRACT CLASS:

A)

CODE:

```
public class abstractclass1{
public static void main(String[] args){
Car c1=new kia();
c1.carName("kia");
c1.carSpeed(122,"kia");
}
}

abstract class Car{
abstract void carName(String name);

abstract void carSpeed(int speed,String name);

abstract void Mileage(int fuel,double mileage);

}
class kia extends Car{
void carName(String name){
System.out.println("Your Car name is "+name);
}
void carSpeed(int speed,String name){
System.out.println(name+" can travel at the speed of "+speed+" km/h");
}
void Mileage(int fuel,double mileage){
System.out.println("kia "+"can travel range of "+(fuel*mileage));
}

}
```

OUTPUT:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac abstractclass1.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java abstractclass1
Your Car name is kia
kia can travel at the speed of 122 km/h
```

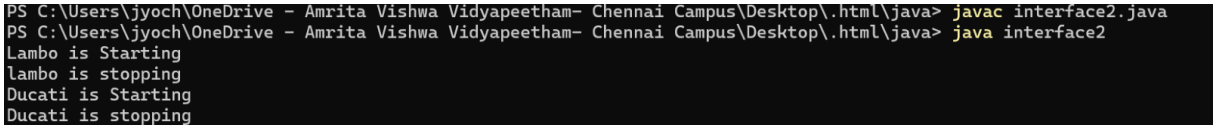
B)

CODE:

```
abstract class Vehicle{

abstract void start(String name);


abstract void stop(String name);


}

class Car extends Vehicle{

void start(String name){

System.out.println( name+" is Starting");

}

void stop(String name){

System.out.println(name+" is stopping");

}

}

class Bike extends Vehicle{

void start(String name){

System.out.println( name+" is Starting");

}

void stop(String name){

System.out.println(name+" is stopping");

}

}

public class interface2{

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

```
Car v1=new Car();


v1.start("Lambo");
v1.stop("lambo");


Bike v2=new Bike();
v2.start("Ducati");
v2.stop("Ducati");
}
}
```

OUTPUT:



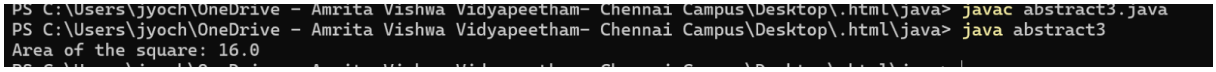
C)

CODE:

```
abstract class Shape {
   abstract double calculateArea();
}


class Square extends Shape {
   private double side;


   public Square(double side) {
      this.side = side;
   }
   double calculateArea() {
      return side * side;
   }
```

```
}
public class abstract3 {
   public static void main(String[] args) {
      Shape square = new Square(4.0);
      System.out.println("Area of the square: " + square.calculateArea());
   }
}
```

OUTPUT:



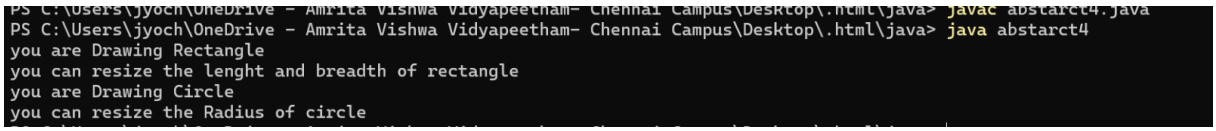
D)

CODE:

```
abstract class Shape2D{
abstract void draw();
abstract void resize();
}
class Rectangle extends Shape2D{
void draw(){
System.out.println("you are Drawing Rectangle");
}
void resize(){
System.out.println("you can resize the lenght and breadth of rectangle");
}
}
class Circle extends Shape2D{
void draw(){
System.out.println("you are Drawing Circle");
}
```

```
void resize(){
System.out.println("you can resize the Radius of circle ");
}
}
public class abstarct4{
public static void main(String[] args){
Shape2D s1=new Rectangle();
s1.draw();
s1.resize();
 Shape2D s2=new Circle();
s2.draw();
s2.resize();
}
}
```

OUTPUT:



13)INTERFACE PROGRAMS:

A)

CODE:

```
interface Shape{ void getArea();
 }
class Rectangle implements Shape
{ double lenght,breadth;
Rectangle(double lenght, double breadth) {
    this.lenght = lenght;
```

```
    this.breadth = breadth;
}
public void getArea(){
   System.out.println("The area of rectangle "+(lenght*breadth));
}


}
```

class Circle implements Shape{ double radius; Circle(double radius){

this.radius=radius;

}

public void getArea()

{ System.out.println("The are of circle is "+(3.14*(radius*radius)));

}

}

public class Interface1 {

public static void main(String[] args) {

Rectangle r1=new Rectangle(12, 12);

r1.getArea();

Circle c1=new Circle(4);

c1.getArea(); }

}

OUTPUT:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac Interface1.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java Interface1
The area of rectangle 144.0
The are of circle is 50.24
```

B)

CODE:

interface Camara{

```
    void takePhoto();
}
interface MusicPlayer{
    void canPlayMusic();
}
class SmartPhone implements Camara,MusicPlayer{
    public void takePhoto(){
        System.out.println("SmartPhone can take photo");
    }
    public void canPlayMusic(){
        System.out.println("SmartPhone can play music");
    }
}
public class Interface22 {
    public static void main(String[] args) {
        SmartPhone s1=new SmartPhone();
        s1.canPlayMusic();
        s1.takePhoto();
    }
}
```

OUTPUT:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac Interface22.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java Interface22
SmartPhone can play music
SmartPhone can take photo
```

C)

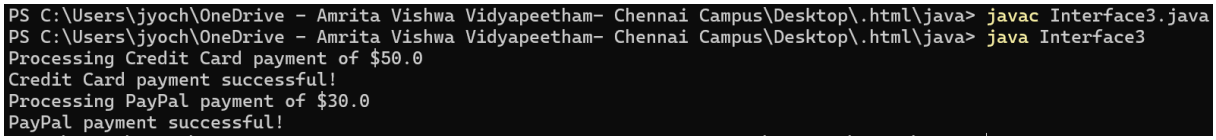
CODE:

```
interface Payment {
    void initiatePayment(double amount);
```

```
    void getPaymentStatus();
}
class CreditCard implements Payment {
    public void initiatePayment(double amount) {
        System.out.println("Processing Credit Card payment of $" + amount);
    }

    public void getPaymentStatus() {
        System.out.println("Credit Card payment successful!");
    }
}

class PayPal implements Payment {
    public void initiatePayment(double amount) {
        System.out.println("Processing PayPal payment of $" + amount);
    }
    public void getPaymentStatus() {
        System.out.println("PayPal payment successful!");
    }
}
public class Interface3 {
    public static void main(String[] args) {
        Payment creditCardPayment = new CreditCard();
        Payment paypalPayment = new PayPal();
         creditCardPayment.initiatePayment(50.0);
        creditCardPayment.getPaymentStatus();
         paypalPayment.initiatePayment(30.0);
        paypalPayment.getPaymentStatus();
```

```
    }
}
```

OUTPUT:



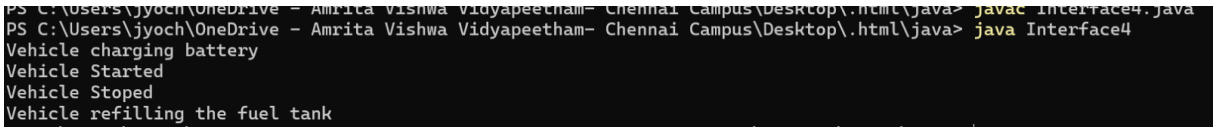
D)

CODE:

```
interface Vehicle{
    void start();
    void stop();
}
interface ElectricVehicle{
    void chargeBattery();
}
interface FuelVehicle{
    void refuel();
}
class HybridCar implements Vehicle,ElectricVehicle,FuelVehicle{
    public void start(){
        System.out.println("Vehicle Started ");
    }
    public void stop(){
        System.out.println("Vehicle Stoped ");
    }
     public void chargeBattery(){
        System.out.println("Vehicle charging battery ");
```

```
    }
     public void refuel(){
        System.out.println("Vehicle refilling the fuel tank ");
    }
 }
public class Interface4 {
    public static void main(String[] args) {
        HybridCar v1=new HybridCar();
        v1.chargeBattery();
        v1.start();
        v1.stop();
        v1.refuel();
    }
}
```

OUTPUT:



14)ENCAPSULATION

A)

CODE:

```
class Person {
    private String name;
    private int age;
    public void setName(String name) {
        this.name = name;
    }
     public String getName() {
```

```
        return name;
    }
     public void setAge(int age) {
        this.age = age;
    }
     public int getAge() {
        return age;
    }
}
 public class encap1 {
    public static void main(String[] args) {
        Person p = new Person();
        p.setName("John");
        p.setAge(20);
        System.out.println("Person: " + p.getName() + ", Age: " + p.getAge());
    }
}
```

OUTPUT:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac encap1.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java encap1
Person: John, Age: 20
```

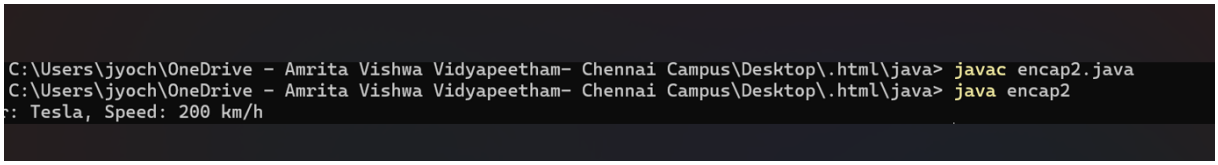
B)

CODE:

```
class Car {
    private String brand;
    private int speed;

    public void setBrand(String brand) {
```

```
        this.brand = brand;
    }

    public String getBrand() {
        return brand;
    }

    public void setSpeed(int speed) {
        if (speed > 200) {
            this.speed = 200;
        } else {
            this.speed = speed;
        }
    }
     public int getSpeed() {
        return speed;
    }
}
 public class encap2 {
    public static void main(String[] args) {
        Car car = new Car();
        car.setBrand("Tesla");
        car.setSpeed(250);
        System.out.println("Car: " + car.getBrand() + ", Speed: " + car.getSpeed() + " km/h");
    }
}
```

OUTPUT:



C)

CODE:

```
class BankAccount {
   private String accountNumber;
   private double balance;

   public BankAccount(String accountNumber, double balance) {
      this.accountNumber = accountNumber;
      this.balance = balance;
   }

   public void deposit(double amount) {
      if (amount > 0) {
         balance += amount;
      }
   }

   public void withdraw(double amount) {
      if (amount > 0 && amount <= balance) {
         balance -= amount;
      }
   }

   public double getBalance() {
      return balance;
   }
}

public class  encap3 {
   public static void main(String[] args) {
      BankAccount account = new BankAccount("12345678", 5000);
      account.deposit(2000);
      account.withdraw(3000);
      System.out.println("Balance: $" + account.getBalance());
   }
}
```

OUTPUT:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac encap3.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java encap3
Balance: $4000.0
```

D)

CODE:

```
import java.lang.*;
class MusicPlayer {
   private String songName;
   private int volume;
    MusicPlayer(String songName) {
      this.songName = songName;
      this.volume = 50;
   }
    String getSongName() {
      return songName;
   }
    void setVolume(int volume) {
      if (volume < 0) this.volume = 0;
      else if (volume > 100) this.volume = 100;
      else this.volume = volume;
   }
    int getVolume() {
      return volume;
   }
}
 public class encap4 {
   public static void main(String[] args) {
      MusicPlayer mp = new MusicPlayer("Shape of You");
      mp.setVolume(120);
```

```
    System.out.println("Now playing: " + mp.getSongName() + ", Volume: " +
mp.getVolume());
  }
}
```

OUTPUT:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac encap4.j
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java encap4
Now playing: Shape of You, Volume: 100
```

15.PACKAGES:

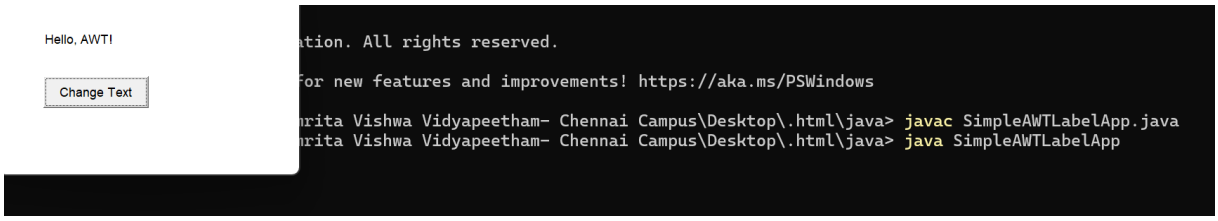
i)

CODE:

```
import java.awt.*;
import java.awt.event.*;

public class SimpleAWTLabelApp {
  SimpleAWTLabelApp() {
    Frame frame = new Frame("AWT Label Example");
    Label label = new Label("Hello, AWT!");
    Button button = new Button("Change Text");
        label.setBounds(50, 50, 150, 30);
    button.setBounds(50, 100, 100, 30);
        button.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        label.setText("Text Updated!");
      }
    });
      frame.add(label);
    frame.add(button);
```

```
        frame.setSize(300, 200);
      frame.setLayout(null);
      frame.setVisible(true);
        frame.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
          frame.dispose();
        }
      });
   }
    public static void main(String[] args) {
      new SimpleAWTLabelApp();
   }
}
```

OUTPUT:



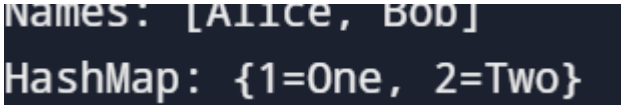
ii)

CODE:

```
import java.util.ArrayList;
import java.util.HashMap;

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

      ArrayList<String> names = new ArrayList<>();
```

```
        names.add("Alice");
        names.add("Bob");
        System.out.println("Names: " + names);
        HashMap<Integer, String> map = new HashMap<>();
        map.put(1, "One");
        map.put(2, "Two");
        System.out.println("HashMap: " + map);
    }
}
```

OUTPUT:



iii)

CODE:

```
package employee;

public class Employee {
     private int empId;
    private String name;
    private double basicSalary;
    private double hra;
    private double da;
    private double tax;
```

```
    public Employee(int empId, String name, double basicSalary, double hra,
double da, double tax) {
        this.empId = empId;
        this.name = name;
        this.basicSalary = basicSalary;
        this.hra = hra;
        this.da = da;
        this.tax = tax;
    }

        public int getEmpId() { return empId; }
    public String getName() { return name; }
    public double getBasicSalary() { return basicSalary; }
    public double getHra() { return hra; }
    public double getDa() { return da; }
    public double getTax() { return tax; }
}
package employee;

public class SalaryCalculator {
    public static double calculateGrossSalary(Employee emp) {
        return emp.getBasicSalary() + emp.getHra() + emp.getDa();
    }

      public static double calculateNetSalary(Employee emp) {
        return calculateGrossSalary(emp) - emp.getTax();
```

```
    }
}
package employee;

public class PayrollDisplay {
    public static void displayPayroll(Employee emp) {
    System.out.println("\n----------------------");
    System.out.println("EMPLOYEE PAYROLL DETAILS");
    System.out.println("----------------------");
    System.out.println("Employee ID   : " + emp.getEmpId());
    System.out.println("Employee Name : " + emp.getName());
    System.out.println("Basic Salary  : ₹" + emp.getBasicSalary());
    System.out.println("HRA           : ₹" + emp.getHra());
    System.out.println("DA            : ₹" + emp.getDa());
    System.out.println("Gross Salary  : ₹" +
SalaryCalculator.calculateGrossSalary(emp));
    System.out.println("Tax Deduction : ₹" + emp.getTax());
    System.out.println("Net Salary    : ₹" +
SalaryCalculator.calculateNetSalary(emp));
    System.out.println("----------------------\n");
    }
}
public class PayrollSystem {
   public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
```

```
        System.out.print("Enter Employee ID: ");
        int empId = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        System.out.print("Enter Employee Name: ");
        String name = scanner.nextLine();

        System.out.print("Enter Basic Salary: ₹");
        double basicSalary = scanner.nextDouble();

        System.out.print("Enter HRA (House Rent Allowance): ₹");
        double hra = scanner.nextDouble();

        System.out.print("Enter DA (Dearness Allowance): ₹");
        double da = scanner.nextDouble();

        System.out.print("Enter Tax Deduction: ₹");
        double tax = scanner.nextDouble();

        Employee emp = new Employee(empId, name, basicSalary, hra, da,
tax);

            PayrollDisplay.displayPayroll(emp);

        scanner.close();
    }
```

}

OUTPUT:

```
 Enter Employee ID: 101
 Enter Employee Name: John Doe
 Enter Basic Salary: ₹50000
 Enter HRA (House Rent Allowance): ₹10000
 Enter DA (Dearness Allowance): ₹5000
 Enter Tax Deduction: ₹8000
```

```
---------------------
EMPLOYEE PAYROLL DETAILS
---------------------
Employee ID   : 101
Employee Name : John Doe
Basic Salary  : ₹50000.0
```

iv)

CODE:

```
package details;


public class Student {
   private int studentId;
   private String name;
   private double marks1, marks2, marks3;


   public Student(int studentId, String name, double marks1, double marks2, double marks3) {
      this.studentId = studentId;
      this.name = name;
      this.marks1 = marks1;
```

```
        this.marks2 = marks2;
        this.marks3 = marks3;
    }

    public int getStudentId() { return studentId; }
    public String getName() { return name; }
    public double getMarks1() { return marks1; }
    public double getMarks2() { return marks2; }
    public double getMarks3() { return marks3; }
}
package details;

public class MarksCalculator {
    public static double calculateTotal(Student stu) {
        return stu.getMarks1() + stu.getMarks2() + stu.getMarks3();
    }

    public static char calculateGrade(double total) {
        double avg = total / 3;
        if (avg >= 90) return 'A';
        else if (avg >= 75) return 'B';
        else if (avg >= 50) return 'C';
        else return 'F';
    }
}
package details;
```

```
public class Display {
    public static void showStudentDetails(Student stu) {
        double total = MarksCalculator.calculateTotal(stu);
        char grade = MarksCalculator.calculateGrade(total);

        System.out.println("\n----------------------");
        System.out.println("STUDENT REPORT CARD");
        System.out.println("----------------------");
        System.out.println("Student ID  : " + stu.getStudentId());
        System.out.println("Student Name: " + stu.getName());
        System.out.println("Total Marks : " + total);
        System.out.println("Grade       : " + grade);
        System.out.println("----------------------\n");
    }
}
import details.Student;
import details.MarksCalculator;
import details.Display;
import java.util.Scanner;

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

        // Taking Student Details
```

```
        System.out.print("Enter Student ID: ");
        int studentId = scanner.nextInt();
        scanner.nextLine();

        System.out.print("Enter Student Name: ");
        String name = scanner.nextLine();

        System.out.print("Enter Marks in Subject 1: ");
        double marks1 = scanner.nextDouble();

        System.out.print("Enter Marks in Subject 2: ");
        double marks2 = scanner.nextDouble();

        System.out.print("Enter Marks in Subject 3: ");
        double marks3 = scanner.nextDouble();

        // Creating Student Object
        Student stu = new Student(studentId, name, marks1, marks2, marks3);

        // Displaying Student Report
        Display.showStudentDetails(stu);
        scanner.close();
    }
}
```

OUTPUT:

```
Enter Student ID: 2001
Enter Student Name: Alice
Enter Marks in Subject 1: 85
Enter Marks in Subject 2: 78
Enter Marks in Subject 3: 90
```

```
----------------------
STUDENT REPORT CARD
----------------------
Student ID  : 2001
Student Name: Alice
Total Marks : 253.0
```

16.EXCEPTION HANDLING

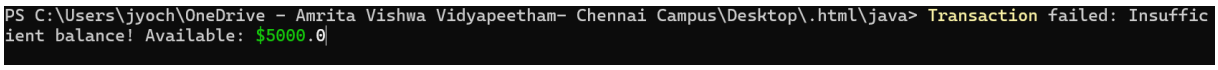
i)

CODE:

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

public class ATM {
   private static double balance = 5000; // Initial balance

   public static void main(String[] args) {
      try {
         withdraw(6000); // Trying to withdraw more than the balance
      } catch (InsufficientFundsException e) {
         System.out.println("Transaction failed: " + e.getMessage());
```

```
    }
  }

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

OUTPUT:



ii)

CODE:

```
class AgeRestrictionException extends Exception {
  AgeRestrictionException(String message) {
    super(message);
  }
}

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

```
        try {
            bookTicket(3); // Booking ticket for a 3-year-old
        } catch (AgeRestrictionException e) {
            System.out.println("Booking failed: " + e.getMessage());
        }
    }

    public static void bookTicket(int age) throws AgeRestrictionException {
        if (age < 5) {
            throw new AgeRestrictionException("Children below 5 years cannot travel alone.");
        }
        System.out.println("Ticket booked successfully!");
    }
}
```

OUTPUT:

```
Booking failed: Children below 5 years cannot travel alone.
```

iii)

CODE:

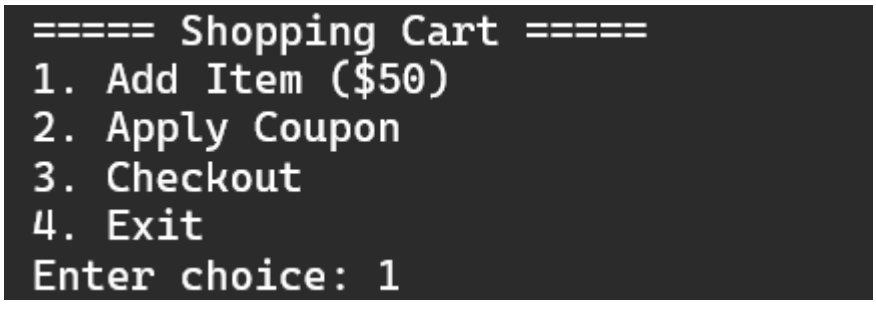
```
import java.util.Scanner;

class InvalidCouponException extends Exception {
    InvalidCouponException(String message) {
        super(message);
    }
}
```

```
class OutOfStockException extends Exception {
    OutOfStockException(String message) {
        super(message);
    }
}

public class ShoppingCart {
    private static double cartTotal = 0;
    private static final String validCoupon = "DISCOUNT20";
    private static boolean productInStock = true; // Simulating product
availability

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

        do {
            System.out.println("\n===== Shopping Cart =====");
            System.out.println("1. Add Item ($50)");
            System.out.println("2. Apply Coupon");
            System.out.println("3. Checkout");
            System.out.println("4. Exit");
            System.out.print("Enter choice: ");
            choice = scanner.nextInt();
```

```
        try {
            switch (choice) {
                case 1:
                    addItem();
                    break;
                case 2:
                    System.out.print("Enter coupon code: ");
                    String coupon = scanner.next();
                    applyCoupon(coupon);
                    break;
                case 3:
                    checkout();
                    break;
                case 4:
                    System.out.println("Thank you for shopping with us!");
                    break;
                default:
                    System.out.println("Invalid choice. Try again.");
            }
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    } while (choice != 4);
}

public static void addItem() throws OutOfStockException {
```

```
        if (!productInStock) throw new OutOfStockException("Product is out of
stock!");
        cartTotal += 50;
        System.out.println("Item added! Cart total: $" + cartTotal);
    }

    public static void applyCoupon(String coupon) throws
InvalidCouponException {
        if (!coupon.equals(validCoupon)) throw new
InvalidCouponException("Invalid coupon code: " + coupon);
        cartTotal *= 0.8;
        System.out.println("Coupon applied! New total: $" + cartTotal);
    }

    public static void checkout() {
        System.out.println("Order placed! Final amount: $" + cartTotal);
    }
}
```

OUTPUT:



```
Enter choice: 1
Item added! Cart total: $50.0

===== Shopping Cart =====
1. Add Item ($50)
2. Apply Coupon
```

```
3. Checkout
4. Exit
Enter choice: 2
Enter coupon code: SAVE50
Error: Invalid coupon code: SAVE50
```

iv)

CODE:

```
import java.util.Scanner;


class AgeRestrictionException extends Exception {
    AgeRestrictionException(String message) {
        super(message);
    }
}


class NoSeatsAvailableException extends Exception {
    NoSeatsAvailableException(String message) {
        super(message);
    }
}


class InvalidInputException extends Exception {
```

```
    InvalidInputException(String message) {
        super(message);
    }
}

public class RailwayBooking {
    private static int availableSeats = 3; // Initial seat availability

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

        do {
            System.out.println("\n===== Railway Ticket Booking =====");
            System.out.println("1. Book Ticket");
            System.out.println("2. Check Available Seats");
            System.out.println("3. Exit");
            System.out.print("Enter choice: ");
            choice = scanner.nextInt();

            try {
                switch (choice) {
                    case 1:
                        System.out.print("Enter Passenger Name: ");
                        String name = scanner.next();
                        System.out.print("Enter Passenger Age: ");
```

```
                int age = scanner.nextInt();
                bookTicket(name, age);
                break;
            case 2:
                System.out.println("Available Seats: " + availableSeats);
                break;
            case 3:
                System.out.println("Thank you for using our Railway Booking System!");
                break;
            default:
                throw new InvalidInputException("Invalid choice! Please enter a valid option.");
            }
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    } while (choice != 3);
  }

  public static void bookTicket(String name, int age) throws AgeRestrictionException, NoSeatsAvailableException {
    if (age < 5) throw new AgeRestrictionException("Children below 5 years cannot travel alone.");
    if (availableSeats == 0) throw new NoSeatsAvailableException("No seats available!");
    availableSeats--;
```

```
        System.out.println(" 🎉 Ticket Booked Successfully! Passenger: " + name +
" | Age: " + age);
        System.out.println("Remaining Seats: " + availableSeats);
    }
}
```

OUTPUT:

```
===== Railway Ticket Booking =====
1. Book Ticket
2. Check Available Seats
3. Exit
Enter choice: 1
Enter Passenger Name: Ramesh
Enter Passenger Age: 30
 🎉 Ticket Booked Successfully! Passenger: Ramesh | Age: 30
Remaining Seats: 2
```

17.FILE HANDLING

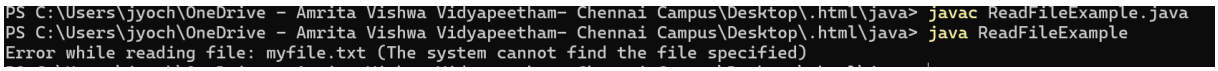
i)

CODE:

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


public class ReadFileExample {
    public static void main(String[] args) {
        try {
            FileReader reader = new FileReader("myfile.txt");
            BufferedReader buffer = new BufferedReader(reader);
            String line = buffer.readLine();
            while (line != null) {
```

```
            System.out.println(line);
            line = buffer.readLine();
        }
        buffer.close();
    } catch (Exception e) {
        System.out.println("Error while reading file: " + e.getMessage());
    }
  }
}
```

OUTPUT:



ii)

CODE:

```
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:

```
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> javac WriteFileExample.java
PS C:\Users\jyoch\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Desktop\.html\java> java WriteFileExample
Successfully wrote to the file.
```

iii)

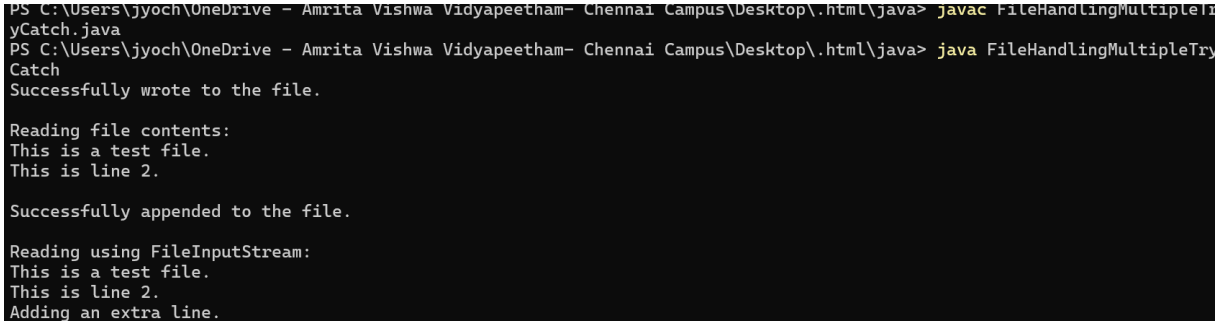
CODE:

```
import java.io.*;


public class FileHandlingMultipleTryCatch {
    public static void main(String[] args) {
        String fileName = "example.txt";
         try (FileWriter writer = new FileWriter(fileName)) {
            writer.write("This is a test file.\n");
            writer.write("This is line 2.");
            System.out.println("Successfully wrote to the file.");
        } catch (IOException e) {
            System.out.println("Error writing file: " + e.getMessage());
        }
```

```
        try (BufferedReader reader = new BufferedReader(new
FileReader(fileName))) {
            System.out.println("\nReading file contents:");
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            System.out.println("Error reading file: " + e.getMessage());
        }
        try (FileWriter writer = new FileWriter(fileName, true)) {
            writer.write("\nAdding an extra line.");
            System.out.println("\nSuccessfully appended to the file.");
        } catch (IOException e) {
            System.out.println("Error appending to file: " + e.getMessage());
        }
        try (FileInputStream fis = new FileInputStream(fileName)) {
            System.out.println("\nReading using FileInputStream:");
            int content;
            while ((content = fis.read()) != -1) {
                System.out.print((char) content);
            }
        } catch (IOException e) {
            System.out.println("Error reading file: " + e.getMessage());
        }
    }
```

}

OUTPUT:



IV)

CODE:

```
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: