SCHOOL OF COMPUTING

LAB RECORD

23CSE111 – Object Oriented Programming

Submitted by

CH.SC.U4CSE24152 – Y.V.S.Likesh

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING

CHENNAI

April - 2025

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING, CHENNAI

BONAFIDE CERTIFICATE

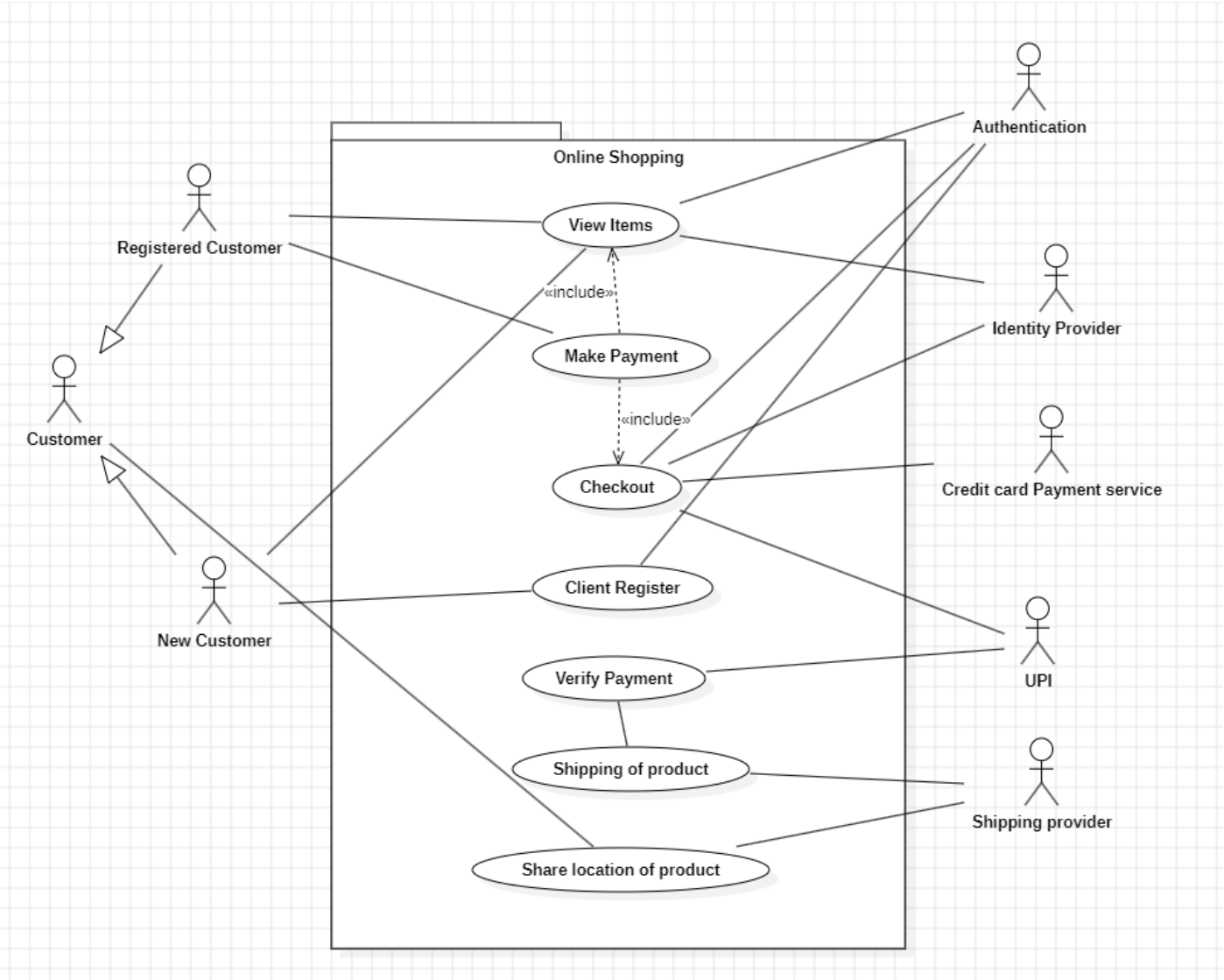
This is to certify that the Lab Record work for 23CSE111- Object Oriented Programming Subject submitted by ***CH.SC.U4CSE24152 – Yerrabolu Venkata Sai Likesh*** in **"Computer Science and Engineering"** is a Bonafide record of the work carried out under my guidance and supervision at Amrita School of Computing, Chennai.

This Lab examination held on 13/03/2025

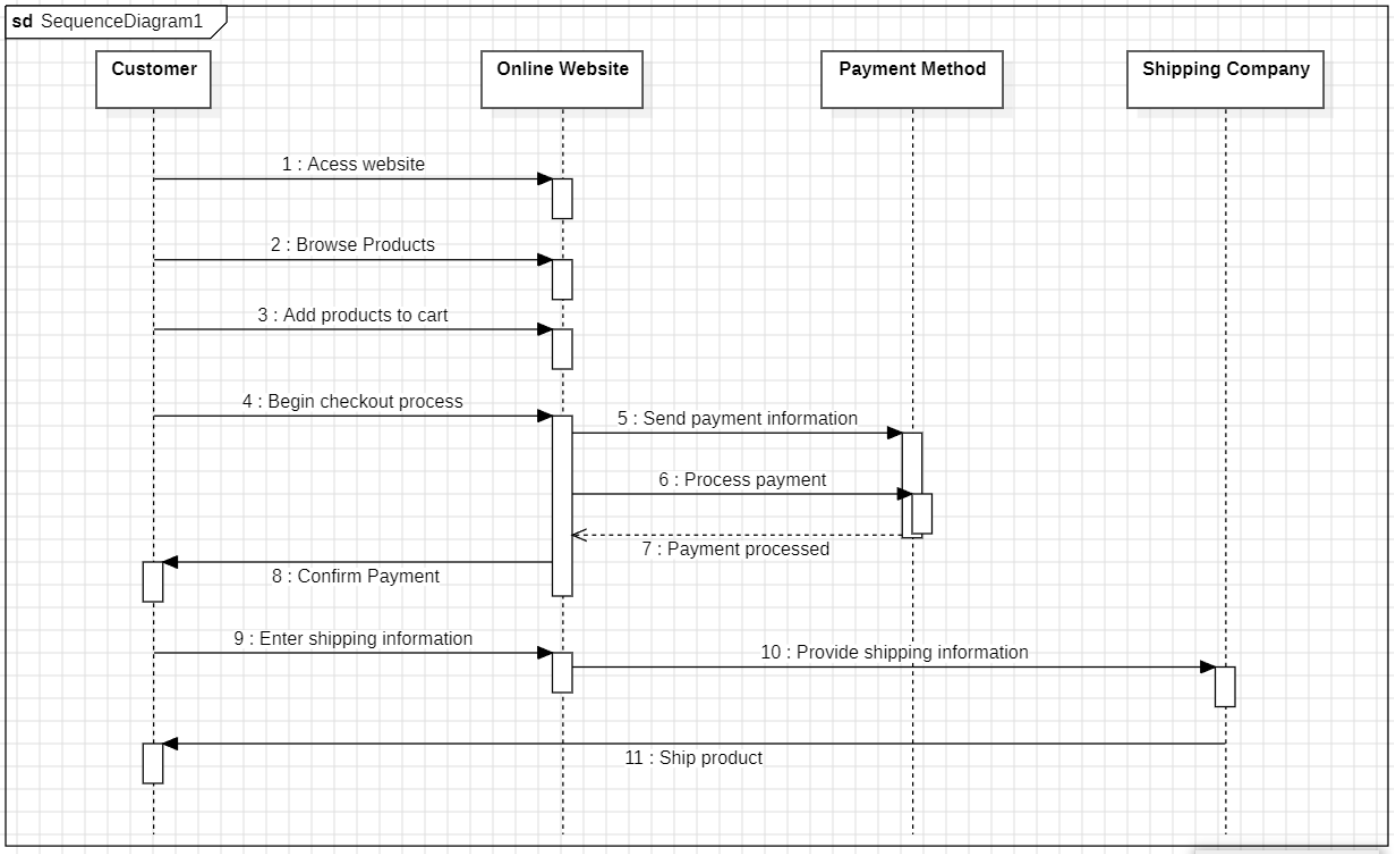
INDEX

EXPERIMENT-1(Online Shopping System)

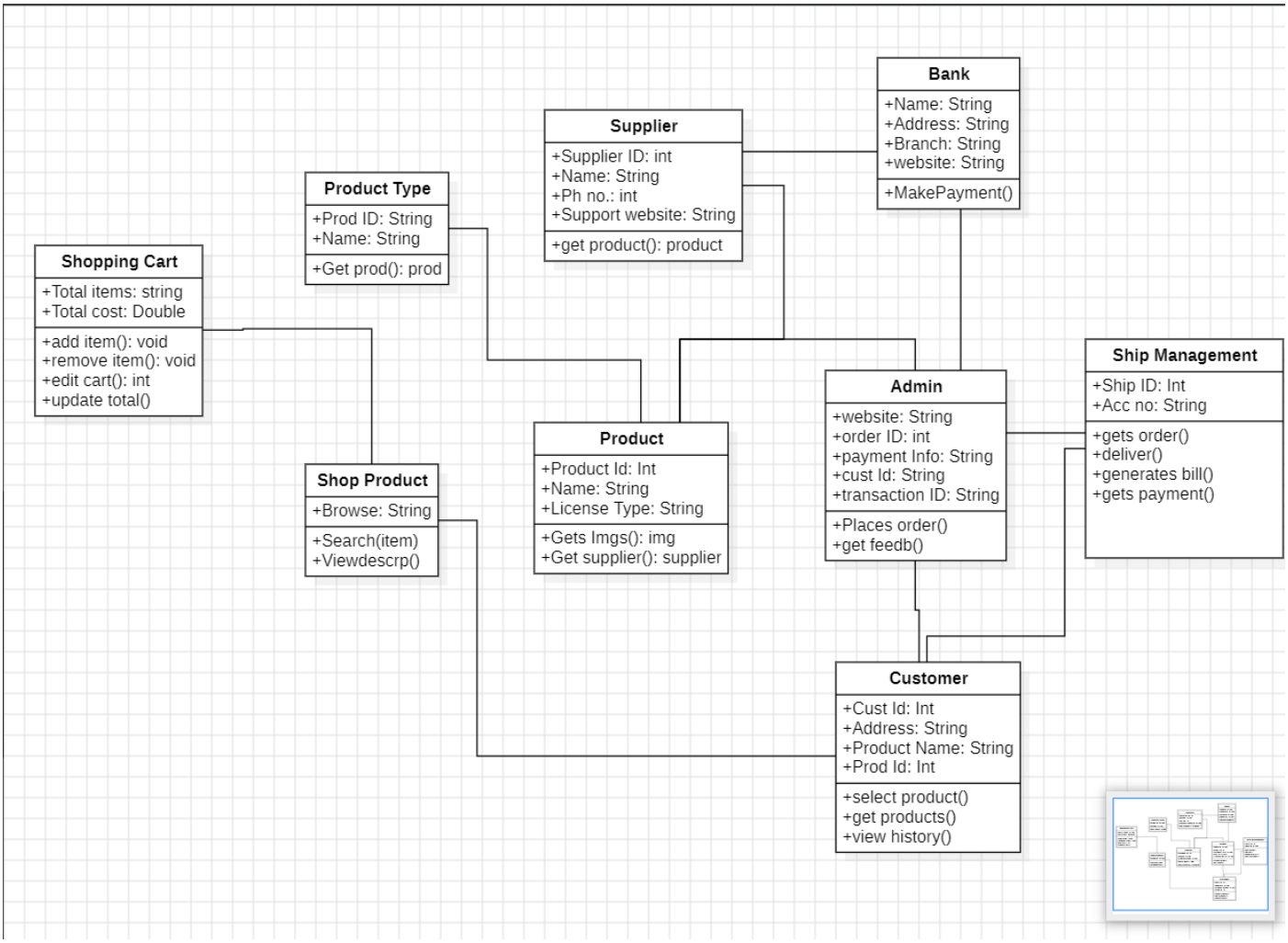
Use Case Diagram:



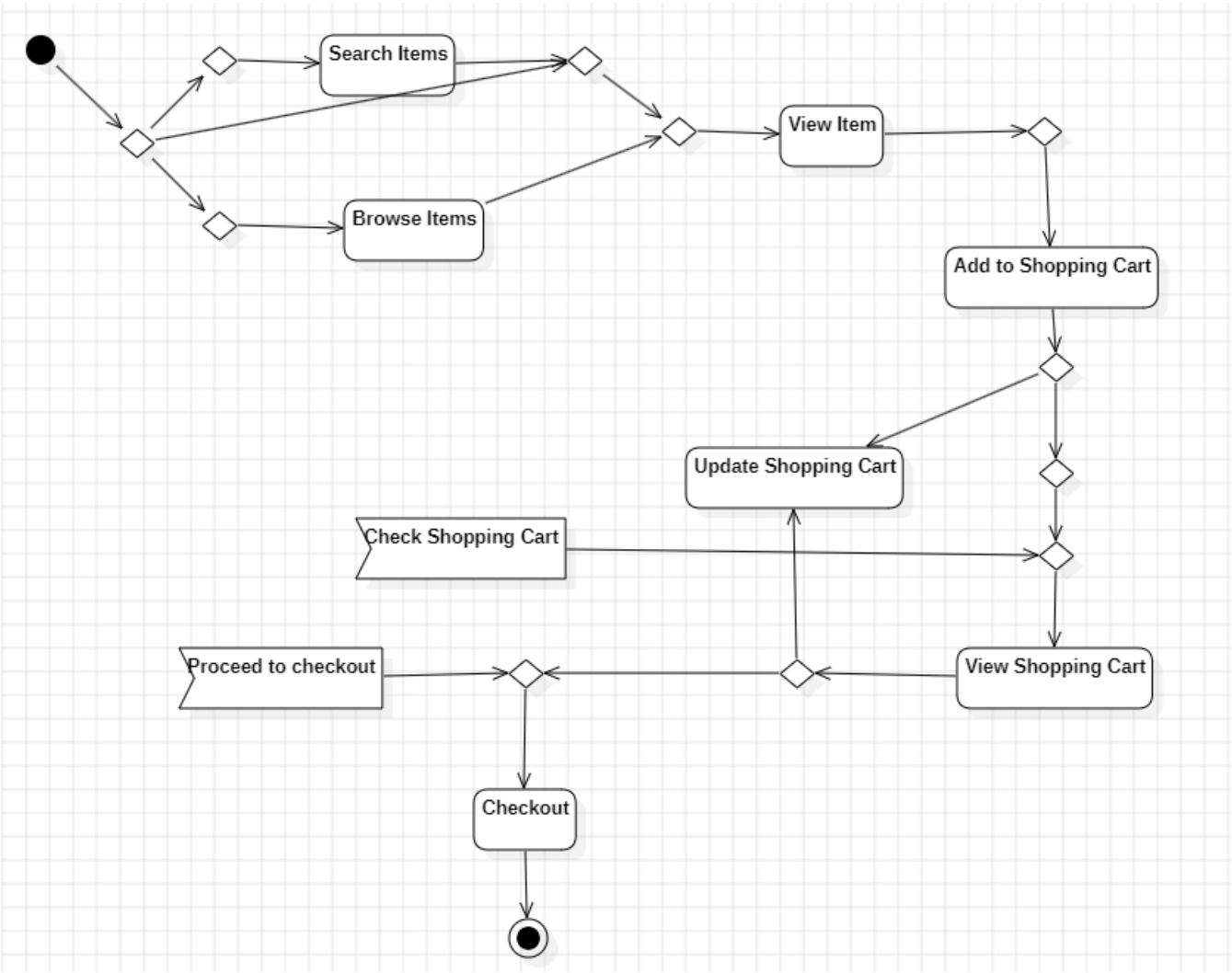
Sequence Diagram:



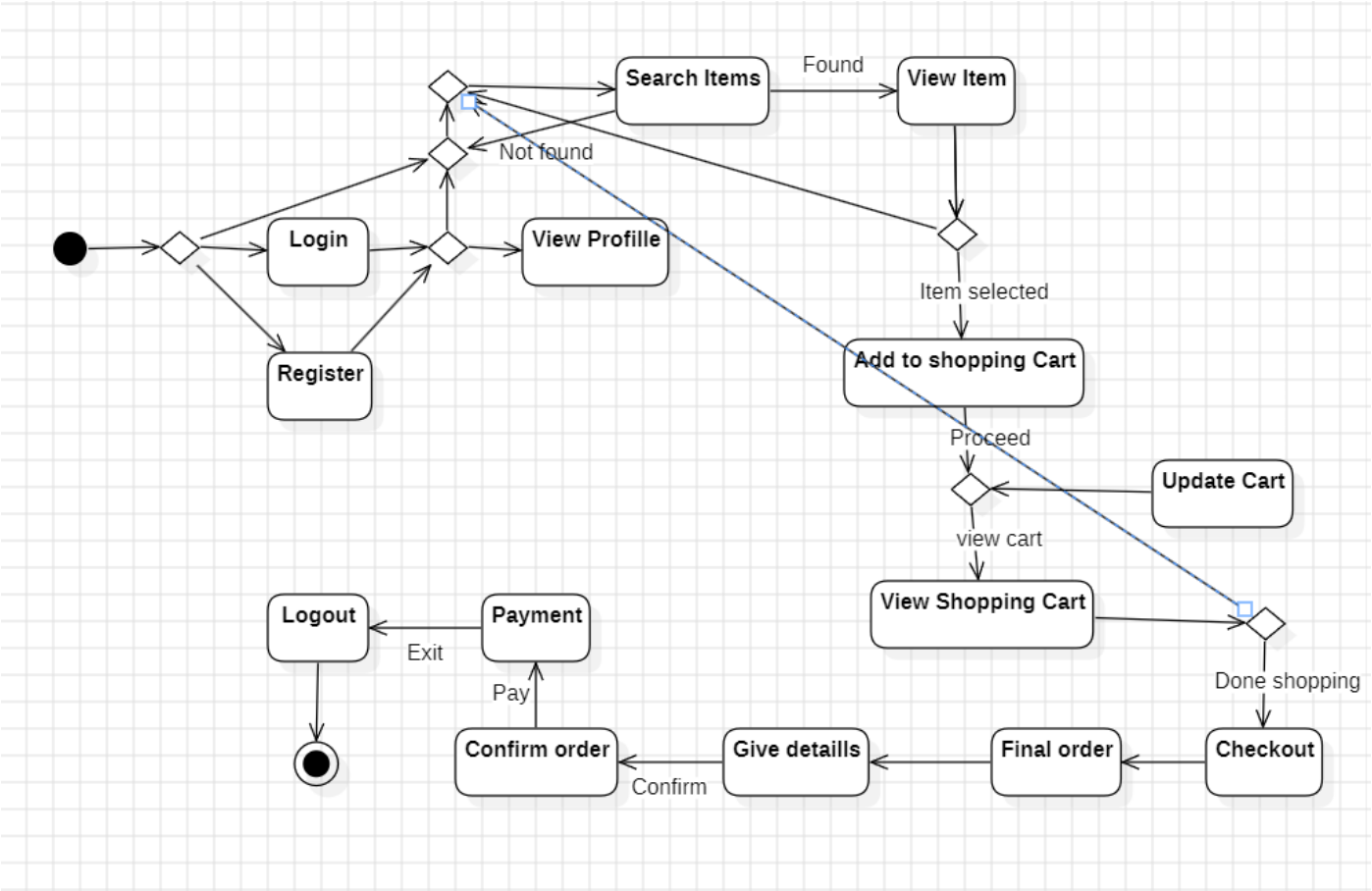
Class Diagram:



Activity Diagram:

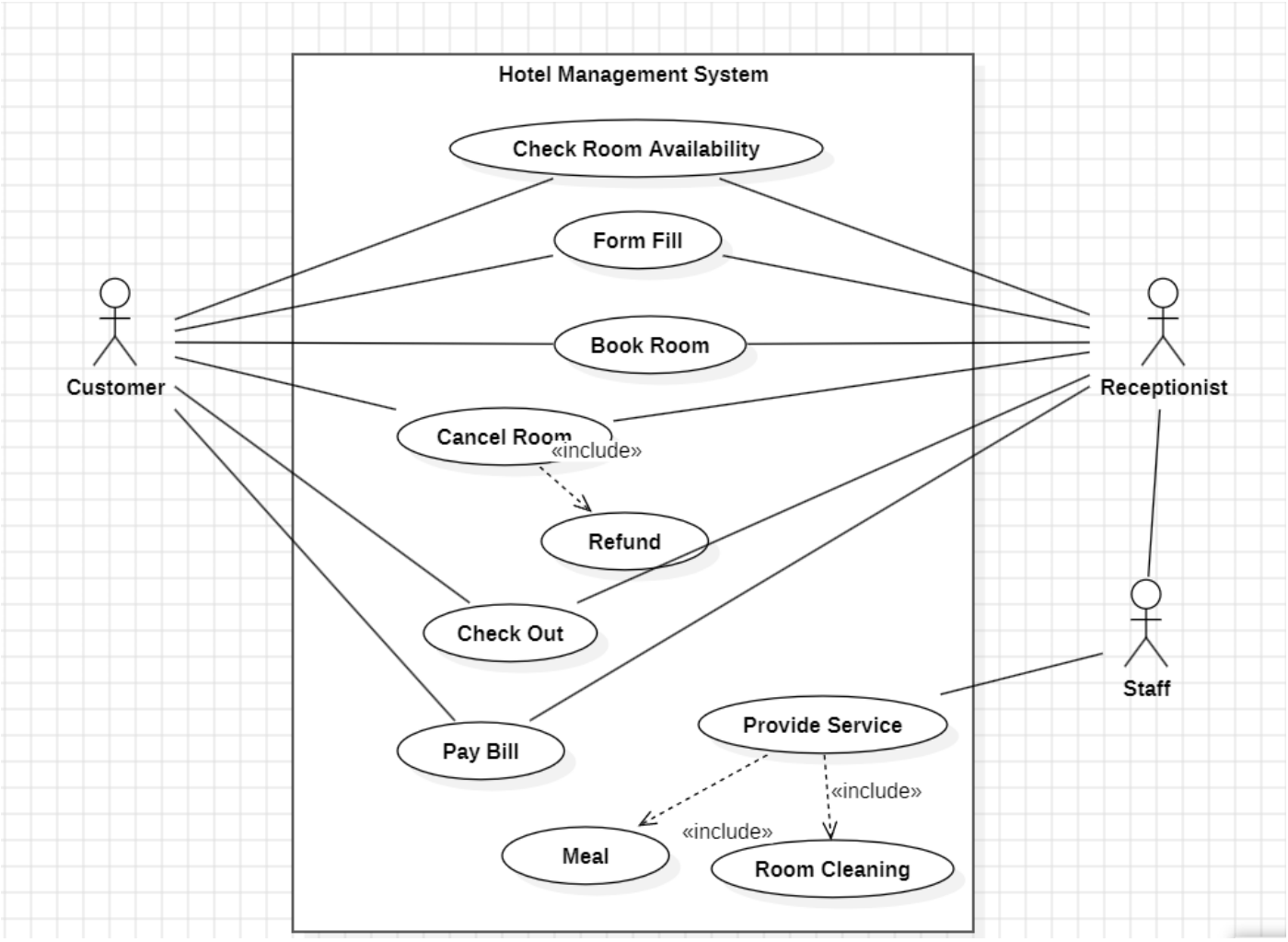


State Diagram:

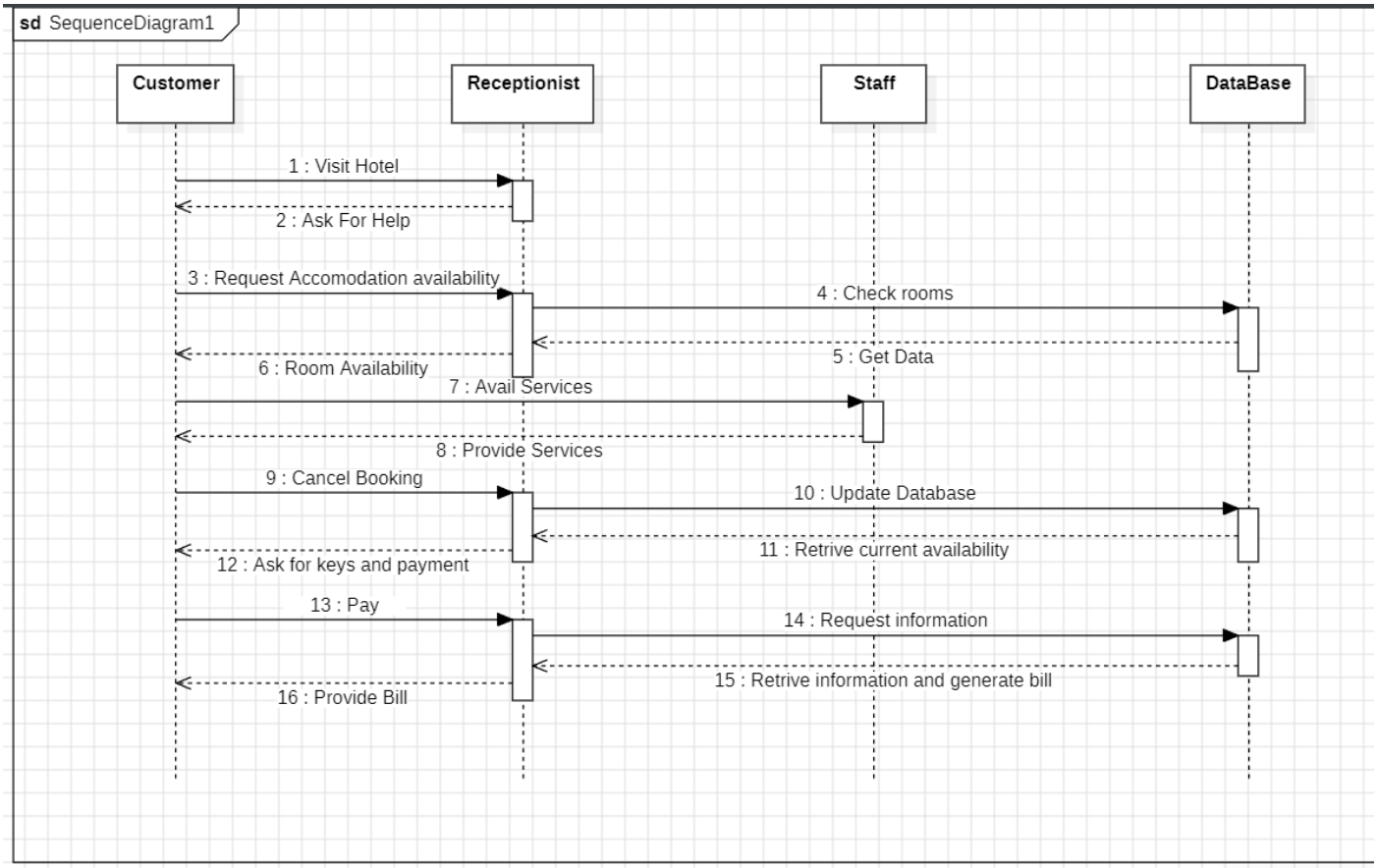


EXPERIMENT-2(Hotel Management System)

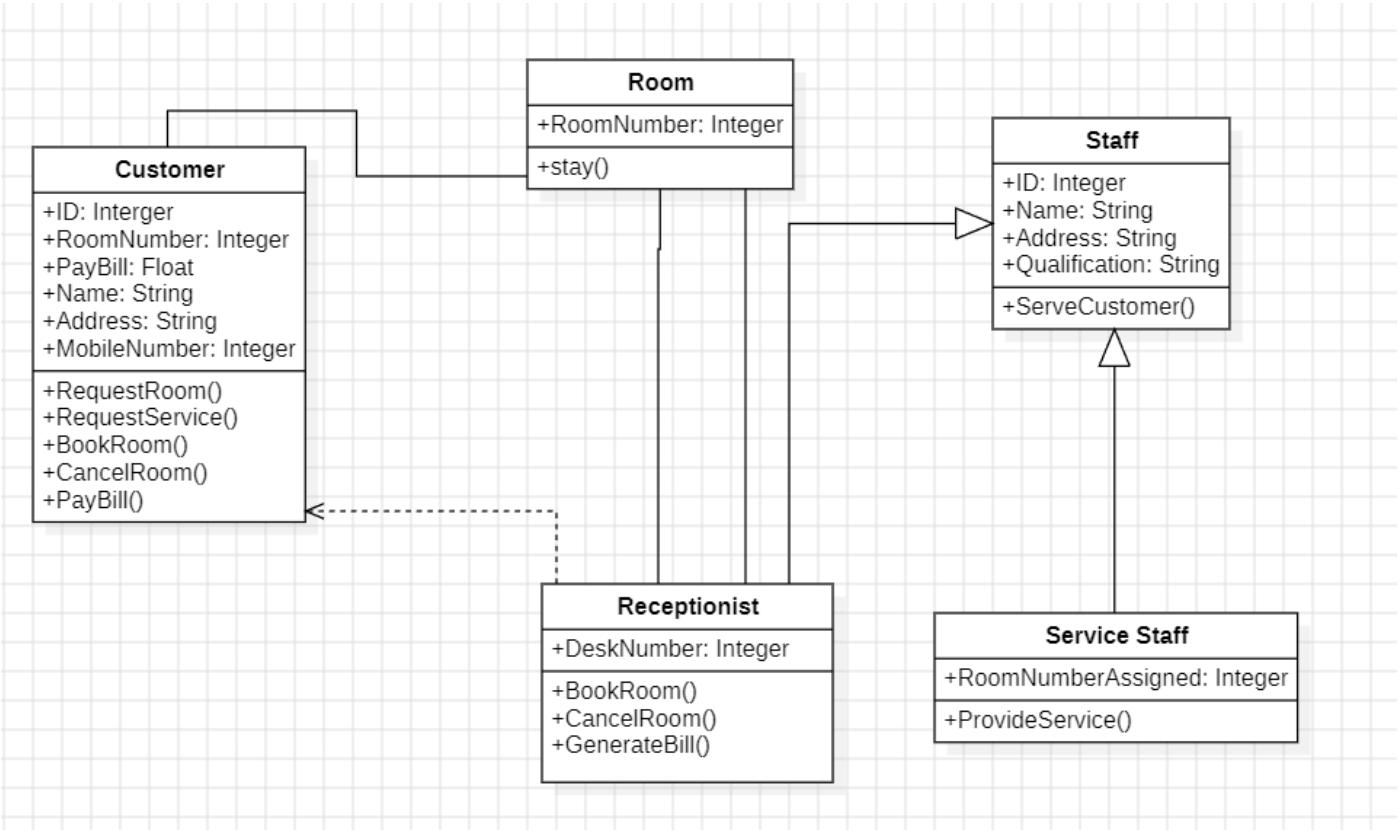
Use Case Diagram:



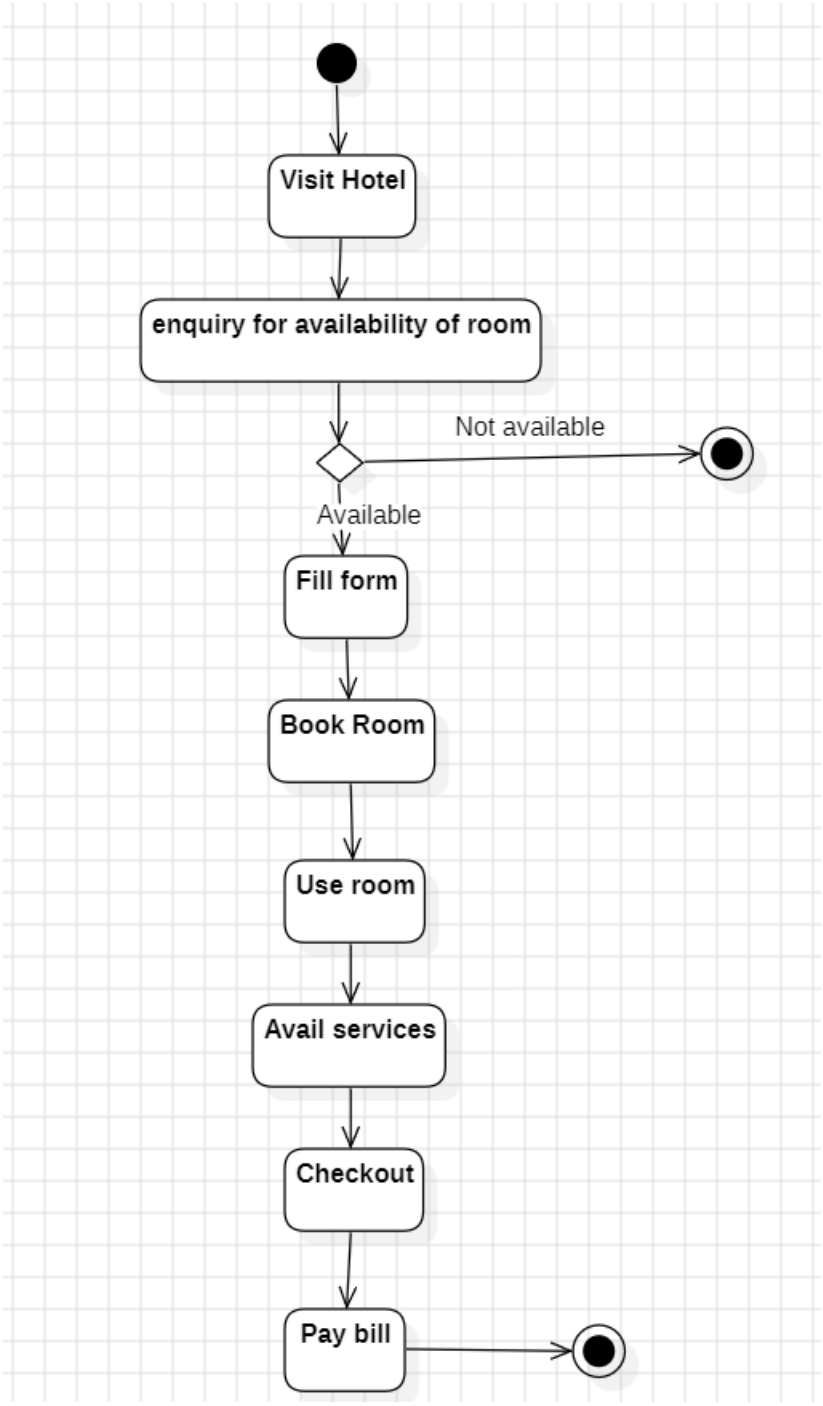
Sequence Diagram:



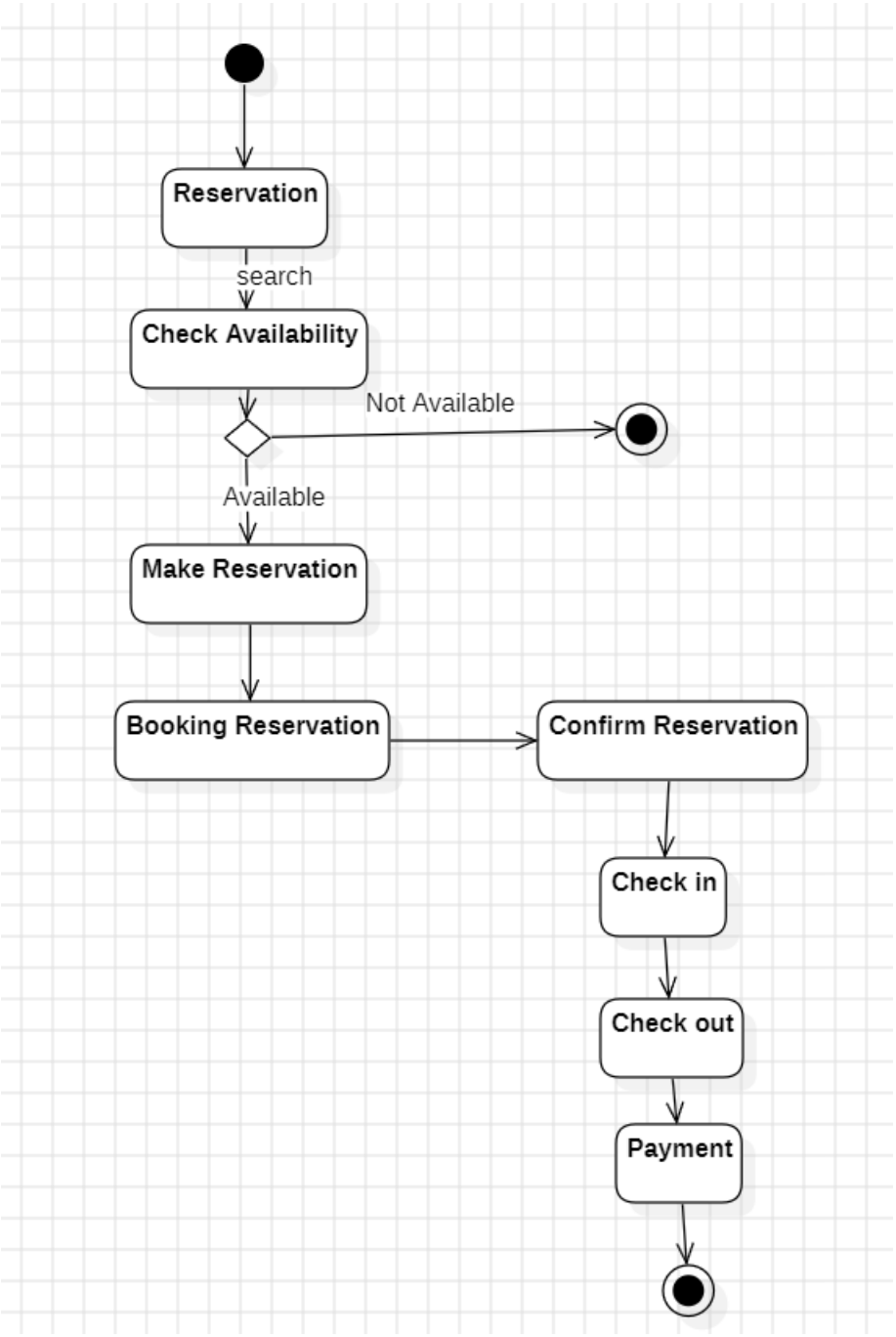
Class Diagram:



Activity Diagram:



State Diagram:



EXPERIMENT-3

Basic Java Codes:

1.Even or Odd:

```java
import java.util.Scanner;
public class EvenOdd{
  public static void main(String[] args){
    Scanner sum = new Scanner(System.in);
    System.out.print("Enter a number: ");
    int num = sum.nextInt();
    if(num % 2 == 0){
        System.out.println(num + " is Even.");
    }else{
        System.out.println(num + " is Odd.");
    }
  }
}
```

Output:

```
D:\Coding\Java Programs>javac EvenOdd.java

D:\Coding\Java Programs>java EvenOdd.java
Enter a number: 5
5 is Odd.
```

2.Reverse Number:

```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) {
        int digit = num % 10;
        rev = rev * 10 + digit;
        num /= 10;
    }
    System.out.println("Reversed Number: " + rev);
  }
}
```

Output:

```
D:\Coding\Java Programs>javac ReverseNumber.java

D:\Coding\Java Programs>java ReverseNumber.java
Enter a number: 123
Reversed Number: 321
```

3.Largest Number:

```java
import java.util.Scanner;
public class LargestNumber{
  public static void main(String[] args){
    Scanner la = new Scanner(System.in);
    System.out.print("Enter first number: ");
    int num1 = la.nextInt();
    System.out.print("Enter second number: ");
    int num2 = la.nextInt();
    System.out.print("Enter third number: ");
    int num3 = la.nextInt();
    int largest = (num1 > num2) ? (num1 > num3 ? num1 : num3) : (num2 > num3 ? num2 : num3);
    System.out.println("Largest number is: " + largest);
  }
}
```

Output:

```
D:\Coding\Java Programs>javac LargestNumber.java

D:\Coding\Java Programs>java LargestNumber.java
Enter first number: 22
Enter second number: 26
Enter third number: 88
Largest number is: 88
```

4.Count Digits:

```java
import java.util.Scanner;
public class CountDigits{
  public static void main(String[] args){
    Scanner co = new Scanner(System.in);
    System.out.print("Enter a number: ");
    int num = co.nextInt();
    int count = 0;
    while(num != 0){
        num /= 10;
        count++;
    }
    System.out.println("Number of digits: " + count);
  }
}
```

Output:

```
D:\Coding\Java Programs>javac CountDigits.java

D:\Coding\Java Programs>java CountDigits.java
Enter a number: 123
Number of digits: 3
```

5.Factorial:

```java
import java.util.Scanner;
public class Factorial{
  public static void main(String[] args){
    Scanner an = new Scanner(System.in);
    System.out.print("Enter a number: ");
    int num = an.nextInt();
    int fact = 1;
    for(int i = 1; i <= num; i++){
        fact *= i;
    }
    System.out.println("Factorial of " + num + " is " + fact);
  }
}
```

Output:

```
D:\Coding\Java Programs>javac Factorial.java

D:\Coding\Java Programs>java Factorial.java
Enter a number: 5
Factorial of 5 is 120
```

6.Fibonaci:

```java
import java.util.Scanner;
public class Fibonacci{
  public static void main(String[] args){
    Scanner my = new Scanner(System.in);
    System.out.print("Enter the number of terms: ");
    int n = my.nextInt();
    int a = 0, b = 1;
    System.out.print("Fibonacci Series: " + a + " " + b);
    for (int i = 2; i < n; i++){
        int next = a + b;
        System.out.print(" " + next);
        a = b;
        b = next;
    }
  }
}
```

Output:

```
D:\Coding\Java Programs>javac Fibonacci.java

D:\Coding\Java Programs>java Fibonacci.java
Enter the number of terms: 5
Fibonacci Series: 0 1 1 2 3
```

7.Prime Check:

```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 <= Math.sqrt(num); 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.");
    }
  }
}
```

Output:

```
D:\Coding\Java Programs>javac PrimeCheck.java

D:\Coding\Java Programs>java PrimeCheck.java
Enter a number: 55
55 is not a Prime Number.
```

8.Swap Numbers:

```java
import java.util.Scanner;
public class SwapNumbers{
  public static void main(String[] args){
    Scanner aa = new Scanner(System.in);
    System.out.print("Enter first number: ");
    int a = aa.nextInt();
    System.out.print("Enter second number: ");
    int b = aa.nextInt();
    a = a + b;
    b = a - b;
    a = a - b;
    System.out.println("After swapping: a = " + a + ", b = " + b);
    }
}
```

Output:

```
D:\Coding\Java Programs>javac SwapNumbers.java

D:\Coding\Java Programs>java SwapNumbers.java
Enter first number: 1234
Enter second number: 12
After swapping: a = 12, b = 1234
```

9.Palindrome:

```java
import java.util.Scanner;
public class Palindrome{
  public static void main(String[] args){
    Scanner aa = new Scanner(System.in);
    System.out.print("Enter a number: ");
    int num = aa.nextInt();
    int original = num, rev = 0;
    while (num != 0){
        int digit = num % 10;
        rev = rev * 10 + digit;
        num /= 10;
    }
    if(original == rev){
        System.out.println(original + " is a Palindrome.");
    }else{
        System.out.println(original + " is not a Palindrome.");
    }
  }
}
```

Output:

```
D:\Coding\Java Programs>javac Palindrome.java

D:\Coding\Java Programs>java Palindrome.java
Enter a number: 1234
1234 is not a Palindrome.
```

10.Armstrong:

```java
import java.util.Scanner;
public class Armstrong{
  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;
    while(num != 0){
        int digit = num % 10;
        sum += digit * digit * digit;
        num /= 10;
    }
    if(sum == original){
        System.out.println(original + " is an Armstrong Number.");
    }else{
        System.out.println(original + " is not an Armstrong Number.");
    }
  }
}
```

Output:

```
D:\Coding\Java Programs>javac Armstrong.java

D:\Coding\Java Programs>java Armstrong.java
Enter a number: 234
234 is not an Armstrong Number.
```

EXPERIMENT-4

Inheritance Codes:

Simple Inheritance:

1.Simple1:

```
class Person {
   String name;
   Person(String name) {
      this.name = name;
   }
   void showName() {
      System.out.println("Name: " + name);
   }
}
class Student extends Person {
   int rollNo;
   Student(String name, int rollNo) {
      super(name);
      this.rollNo = rollNo;
   }
   void showRollNo() {
      System.out.println("Roll No: " + rollNo);
   }
}
public class Simple1{
   public static void main(String[] args) {
      Student s = new Student("Likesh", 101);
```

```
        s.showName();
        s.showRollNo();
    }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Simple Inheritance>javac Simple1.java

C:\Rahi Pics\Java Programs\Simple Inheritance>java Simple1
Name: Likesh
Roll No: 101
```

2.Simple2:

```
class Animal {
    void eat() {
        System.out.println("This animal eats food.");
    }
}
class Dog extends Animal {
    void bark() {
        System.out.println("The dog barks.");
    }
}
public class Simple2{
    public static void main(String[] args) {
        Dog d = new Dog();
        d.eat();
        d.bark();
    }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Simple Inheritance>javac Simple2.java

C:\Rahi Pics\Java Programs\Simple Inheritance>java Simple2
This animal eats food.
The dog barks.
```

Multilevel Inheritance:

1.Dog:

```
class Dog{
  void eat(){
    System.out.println("Dog is eating");
  }
}
class Puppy extends Dog{
  void bark(){
    System.out.println("Dog is barking");
  }
}
class Labrador extends Puppy{
  void display(){
    System.out.println("Labrador is a type of Dog");
  }
}
public class Main{
  public static void main(String[] args){
    Labrador labrador = new Labrador();
    labrador.eat();
```

```
    labrador.bark();
    labrador.display();
  }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Mutilevel Inheritnce>javac Dog.java

C:\Rahi Pics\Java Programs\Mutilevel Inheritnce>java Dog
Dog is eating
Dog is barking
Labrador is a type of Dog
```

2.Object:

```
class Object{
  public void display(){
    System.out.println("Inside display");
  }
}
class Rectangle extends Object{
  public void area() {
     System.out.println("Inside area");
  }
}
class Cube extends Rectangle{
  public void volume(){
    System.out.println("Inside volume");
  }
}
public class Tester{
```

```
 public static void main(String[] arguments){
   Cube cube = new Cube();
   cube.display();
   cube.area();
   cube.volume();
 }
}
```

Output:

```
C:\Rahi Pics\Java Programs\Mutilevel Inheritnce>javac Object.java

C:\Rahi Pics\Java Programs\Mutilevel Inheritnce>java Object
Inside display
Inside area
Inside volume
```

Heirarchical Inheritance:

1.Hirec1:

```
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 Square extends Shape {
   void area() {
      System.out.println("Area of square = side * side");
   }
}
public class Hirec1 {
   public static void main(String[] args) {
      Circle circle = new Circle();
      circle.display();
      circle.area();
   }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Hierarchical Inheritance>javac Hirec1.java

C:\Rahi Pics\Java Programs\Hierarchical Inheritance>java Hirec1
This is a shape.
Area of circle = π * r * r
```

2.Hirec2:

```
class Animal {
   void eat() {
      System.out.println("Animal eats food.");
   }
}
class Dog extends Animal {
   void bark() {
      System.out.println("Dog barks.");
   }
}
class Cat extends Animal {
   void meow() {
      System.out.println("Cat meows.");
   }
}
public class Hirec2 {
   public static void main(String[] args) {
      Dog d = new Dog();
      d.eat();
      d.bark();
      Cat c = new Cat();
      c.eat();
```

```
    c.meow();
  }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Hierarchical Inheritance>javac Hirec2.java

C:\Rahi Pics\Java Programs\Hierarchical Inheritance>java Hirec2
Animal eats food.
Dog barks.
Animal eats food.
Cat meows.
```

Hybrid Inheritance:

1.Hybrid1:

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

```
        System.out.println("Car brake applied.");
    }
    void drive() {
        System.out.println("Car is driving...");
    }
}
public class Hybrid1 {
    public static void main(String[] args) {
        Car car = new Car();
        car.fuel();
        car.start();
        car.applyBrake();
        car.drive();
    }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Hierarchical Inheritance>javac Hybrid1.java

C:\Rahi Pics\Java Programs\Hierarchical Inheritance>java Hybrid1
Vehicle uses fuel.
Car engine started.
Car brake applied.
Car is driving...
```

2.Hybrid2:

```
interface A {
    void methodA();
}
interface B {
    void methodB();
```

```
}
class C implements A {
   public void methodA() {
      System.out.println("Method A from Interface A");
   }
}
class D extends C implements B {
   public void methodB() {
      System.out.println("Method B from Interface B");
   }
   void methodD() {
      System.out.println("Method D from class D");
   }
}
public class Hybrid2 {
   public static void main(String[] args) {
      D obj = new D();
      obj.methodA();
      obj.methodB();
      obj.methodD();
   }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Hierarchical Inheritance>javac Hybrid2.java

C:\Rahi Pics\Java Programs\Hierarchical Inheritance>java Hybrid2
Method A from Interface A
Method B from Interface B
Method D from class D
```

EXPERIMENT-5

Polymorphism Codes:

Constructor:

Constructor:

```
class Student {
   String name;
   int age;
Student(String n, int a) {
      name = n;
      age = a;
   }
void display() {
      System.out.println("Name: " + name);
      System.out.println("Age: " + age);
   }
}
public class Constructor{
   public static void main(String[] args) {
      Student stu = new Student("Likesh", 17);
      stu.display();
   }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Constructor>javac Constructor.java

C:\Rahi Pics\Java Programs\Constructor>java Constructor
Name: Likesh
Age: 17
```

CONSTRUCTOR OVERLOADING:

Car:

```
class Car {
String model;
int year;
Car() {
  model = "Unknown";
  year = 0;
}
Car(String model, int year) {
  this.model = model;
  this.year = year;
}
void display() {
  System.out.println("Model: " + model + ", Year: " + year);
}
public static void main(String[] args) {
  Car car1 = new Car();
  Car car2 = new Car("Rolls Royles", 2025);
  car1.display();
  car2.display();
 }
}
```

OUTPUT:

```
C:\Users\CH.SC.U4CSE24152>javac Car.java

C:\Users\CH.SC.U4CSE24152>java Car
Model: Unknown, Year: 0
Model: Rolls Royles, Year: 2025
```

METHOD OVERLOADING

1.Adding:

```
class Adding{
private static void display(int a){
  System.out.println("Arguments: " + a);
}
private static void display(int a, int b){
  System.out.println("Arguments: " + a + " and " + b);
}
public static void main(String[] args){
  display(1);
  display(1, 4);
 }
}
```

OUTPUT:

```
C:\Users\CH.SC.U4CSE24152>javac Adding.java

C:\Users\CH.SC.U4CSE24152>java Adding
Arguments: 1
Arguments: 1 and 4
```

2.Numberfinder:

```
class Numberfinder{
private String formatNumber(int value) {
  return String.format("%d", value);
}
private String formatNumber(double value) {
 return String.format("%.3f", value);
}
private String formatNumber(String value) {
  return String.format("%.2f", Double.parseDouble(value));
}
public static void main(String[] args) {
   Numberfinder hs = new Numberfinder();
   System.out.println(hs.formatNumber(50));
   System.out.println(hs.formatNumber(52.8965));
   System.out.println(hs.formatNumber("50"));
  }
}
```

OUTPUT:

```
C:\Users\CH.SC.U4CSE24152>javac Numberfinder.java

C:\Users\CH.SC.U4CSE24152>java Numberfinder
50
52.897
50.00
```

METHOD OVERRIDING:

1.Boy:

```
class Male
{
public void eat()
{
System.out.println("Male is eating");
}
}
class Boy extends Male
{
public void eat()
{
System.out.println("Boy is eating");
 }
public static void main( String[] args) {
  Male obj = new Boy();
  obj.eat();
 }
}
```

OUTPUT:

```
C:\Users\CH.SC.U4CSE24152>javac Boy.java

C:\Users\CH.SC.U4CSE24152>java Boy
Boy is eating
```

2.Win:

```
class Quadrilateral {
public void display()
{
  System.out.println("I am a Quadrilateral");
}
}
class Rectangle extends Quadrilateral {
public void display()
{
  System.out.println("I am a Rectangle");
}
}
class Win {
  public static void main(String[] args)
  {
    Quadrilateral r = new Rectangle();
    r.display();
  }
}
```

OUTPUT:

```
C:\Users\CH.SC.U4CSE24152>javac Win.java

C:\Users\CH.SC.U4CSE24152>java Win
I am a Rectangle
```

EXPERIMENT-5

Abstraction Codes:

Interface:

1.Interface1:

```
interface Animal{
  void makeSound();
}
class Interface1 implements Animal{
  public void makeSound(){
    System.out.println("Woof!");
  }
  public static void main(String[] args){
    Interface1 dog = new Interface1();
    dog.makeSound();
  }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Interface>javac Interface1.java

C:\Rahi Pics\Java Programs\Interface>java Interface1
Woof!
```

2.Interface2:

```
interface Shape{
  double area();
}
class Circle implements Shape{
  double radius;
  Circle(double r){
    radius = r;
  }
  public double area(){
    return Math.PI * radius * radius;
  }
}
class Square implements Shape{
  double side;
  Square(double s){
    side = s;
  }
  public double area(){
    return side * side;
  }
}
class Interface2{
  public static void main(String[] args){
    Shape circle = new Circle(4);
    Shape square = new Square(8);
    System.out.println("Circle area: " + circle.area());
```

```
    System.out.println("Square area: " + square.area());
  }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Interface>javac Interface2.java

C:\Rahi Pics\Java Programs\Interface>java Interface2
Circle area: 50.26548245743669
Square area: 64.0
```

3.Interface3:

```
interface Vehicle{
  void drive();
  void fill();
}
class Tata implements Vehicle{
  public void drive(){
    System.out.println("Driving silently...");
  }
  public void fill(){
    System.out.println("Keep an eye on petrol...");
  }
}
class Interface3{
  public static void main(String[] args){
    Tata car = new Tata();
    car.drive();
    car.fill();
  }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Interface>javac Interface3.java

C:\Rahi Pics\Java Programs\Interface>java Interface3
Driving silently...
Keep an eye on petrol...
```

4.Interface4:

```
interface Calculator{
  int add(int a, int b);
  default int subtract(int a, int b){
    return a - b;
  }
  static int multiply(int a, int b){
    return a * b;
  }
}
class SimpleCalculator implements Calculator{
  public int add(int a, int b){
    return a + b;
  }
}
class Interface4{
  public static void main(String[] args){
    SimpleCalculator calc = new SimpleCalculator();
    System.out.println("Addition value is: " + calc.add(13, 6));
    System.out.println("Subtraction value is: " + calc.subtract(15, 4));
    System.out.println("Multiplication value is: " + Calculator.multiply(16, 15));
  }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Interface>javac Interface4.java

C:\Rahi Pics\Java Programs\Interface>java Interface4
Addition value is: 19
Subtraction value is: 11
Multiplication value is: 240
```

Abstract Class

1.Abstract1:

```
abstract class Employee{
 String name;
 Employee(String name){
  this.name = name;
 }
 abstract double calculateSalary();
}
class FullTimeEmployee extends Employee{
 double monthlySalary;
 FullTimeEmployee(String name, double salary){
  super(name);
  this.monthlySalary = salary;
 }
 public double calculateSalary(){
  return monthlySalary;
 }
}
class Abstract1{
 public static void main(String[] args){
  FullTimeEmployee emp = new FullTimeEmployee("Mukesh", 50000);
  System.out.println("Employee Name: " + emp.name);
```

```
    System.out.println("Monthly Salary: " + emp.calculateSalary()+"Rs");
  }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Abstract>javac Abstract1.java

C:\Rahi Pics\Java Programs\Abstract>java Abstract1
Employee Name: Mukesh
Monthly Salary: 50000.0Rs
```

2.Abstract2:

```
abstract class Shape{
  abstract double area();
}
class Rectangle extends Shape{
  double len, wid;
  Rectangle(double l, double w){
    len = l;
    wid = w;
  }
  double area(){
    return len * wid;
  }
}
class Triangle extends Shape{
  double bae, hei;
  Triangle(double b, double h){
    bae = b;
    hei = h;
  }
  double area(){
    return (0.5) * bae * hei;
  }
}
class Abstract2{
  public static void main(String[] args){
    Shape rect = new Rectangle(10, 5);
```

```
    Shape tri = new Triangle(8, 6);
    System.out.println("Rectangle Area: " + rect.area());
    System.out.println("Triangle Area: " + tri.area());
  }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Abstract>javac Abstract2.java

C:\Rahi Pics\Java Programs\Abstract>java Abstract2
Rectangle Area: 50.0
Triangle Area: 24.0
```

3.Abstract3:

```
class BankAccount{
 private double balance;
 public void deposit(double amount){
  if (amount > 0)
   balance += amount;
 }
 public void withdraw(double amount){
  if (amount > 0 && amount <= balance)
   balance -= amount;
  else
   System.out.println("Insufficient balance or invalid amount.");
 }
 public double getBalance(){
  return balance;
 }
}
class Abstract3{
 public static void main(String[] args){
  BankAccount Acc = new BankAccount();
  Acc.deposit(1000);
  Acc.withdraw(300);
  Acc.withdraw(800);
  System.out.println("Final Balance: Rs." + Acc.getBalance());
 }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Abstract>javac Abstract3.java

C:\Rahi Pics\Java Programs\Abstract>java Abstract3
Insufficient balance or invalid amount.
Final Balance: Rs.700.0
```

4.Abstract4:

```
abstract class Vehicle{
  String brand;
  Vehicle(String brand){
    this.brand = brand;
  }
  abstract void startEngine();
  void showBrand(){
    System.out.println("Brand: " + brand);
  }
}
class Car extends Vehicle{
  Car(String brand){
    super(brand);
  }
  void startEngine(){
    System.out.println(brand + " engine started.");
  }
}
class Bike extends Vehicle{
  Bike(String brand){
    super(brand);
  }
  void startEngine(){
    System.out.println(brand + " engine started.");
  }
}
```

```
public class Abstract4{
  public static void main(String[] args){
    Vehicle ve1 = new Car("Toyota");
    Vehicle ve2 = new Bike("Duke");
    ve1.showBrand();
    ve1.startEngine();
    ve2.showBrand();
    ve2.startEngine();
  }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Abstract>javac Abstract4.java

C:\Rahi Pics\Java Programs\Abstract>java Abstract4
Brand: Toyota
Toyota engine started.
Brand: Duke
Duke engine started.
```

EXPERIMENT-5

Encapsulation Codes:

Encapsulation:

1.Encapsulation1:

```
class Student{
  private String name;
  private int age;
  public void setName(String name){
    this.name = name;
  }
  public void setAge(int age){
    this.age = age;
  }
  public String getName(){
    return name;
  }
  public int getAge(){
    return age;
  }
}
public class Encapsulation1{
  public static void main(String[] args){
    Student stu = new Student();
    stu.setName("Mohan");
    stu.setAge(20);
    System.out.println("Student's Name is " + stu.getName());
```

```
    System.out.println("Student's Age is " + stu.getAge());
  }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Encapsulation>javac Encapsulation1.java

C:\Rahi Pics\Java Programs\Encapsulation>java Encapsulation1
Student's Name is Mohan
Student's Age is 20
```

2.Encapsulation2:

```
class Person{
 private int age;
 public void setAge(int age){
  if(age > 0 && age <= 100){
   this.age = age;
  }
 else if(age > 100){
   System.out.println("Too Old");
  }
 else{
   System.out.println("Invalid age!");
  }
 }
 public int getAge(){
  return age;
 }
}
public class Encapsulation2{
 public static void main(String[] args){
  Person p = new Person();
  p.setAge(45);
  System.out.println("Age: " + p.getAge());
  p.setAge(-2);
  p.setAge(101);
 }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Encapsulation>javac Encapsulation2.java

C:\Rahi Pics\Java Programs\Encapsulation>java Encapsulation2
Age: 45
Invalid age!
Too Old
```

3.Encapsulation3:

```
class Employee{
  private int id;
  private String nam;
  private double salary;
  public void setDetails(int id, String nam, double salary){
    this.id = id;
    this.nam = nam;
    this.salary = salary;
  }
  public void displayDetails(){
    System.out.println("ID: " + id);
    System.out.println("Name: " + nam);
    System.out.println("Salary: Rs." + salary);
  }
}
public class Encapsulation3{
  public static void main(String[] args){
    Employee emp = new Employee();
    emp.setDetails(101346, "Rahul", 55500);
    emp.displayDetails();
  }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Encapsulation>javac Encapsulation3.java

C:\Rahi Pics\Java Programs\Encapsulation>java Encapsulation3
ID: 101346
Name: Rahul
Salary: Rs.55500.0
```

4.Encapsulation4:

```
class BankAccount{
 private double balance;
 public void deposit(double amount){
  if (amount > 0)
   balance += amount;
  else
   System.out.println("Deposit amount must be positive.");
 }
 public void withdraw(double amount){
  if (amount > 0 && amount <= balance){
   balance -= amount;
   if (balance < 1000){
    System.out.println("Warning: Low balance.");
   }
  }
 else{
   System.out.println("Insufficient balance or invalid amount.");
  }
 }
 public double getBalance(){
  return balance;
 }
}
```

```
public class Encapsulation4{
  public static void main(String[] args){
    BankAccount acco = new BankAccount();
    acco.deposit(10000);
    acco.withdraw(4500);
    System.out.printf("Current Balance: Rs %.2f\n", acco.getBalance());
  }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Encapsulation>javac Encapsulation4.java

C:\Rahi Pics\Java Programs\Encapsulation>java Encapsulation4
Current Balance: Rs 5500.00
```

Packages:

Built-in Packages:

1.Packages1:

```
import java.time.LocalDate;
import java.time.LocalTime;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
public class Packages1{
  public static void main(String[] args){
    LocalDate date = LocalDate.now();
    LocalTime time = LocalTime.now();
    LocalDateTime dateTime = LocalDateTime.now();
    DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-yyyy
HH:mm:ss");
    System.out.println("Current Date: " + date);
    System.out.println("Current Time: " + time);
    System.out.println("Current Date and Time: " + dateTime.format(formatter));
  }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Packages>javac Packages1.java

C:\Rahi Pics\Java Programs\Packages>java Packages1
Current Date: 2025-04-05
Current Time: 19:56:34.785596100
Current Date and Time: 05-04-2025 19:56:34
```

2.Packages2:

```
public class Packages2{
 public static void main(String[] args){
  double a = 9.0;
  double b = -10.5;
  System.out.println("Square root of " + a + ": " + Math.sqrt(a));
  System.out.println("Absolute value of " + b + ": " + Math.abs(b));
  System.out.println("Power (3^5): " + Math.pow(3,5));
  System.out.println("Maximum of 100 and 120: " + Math.max(100, 120));
  System.out.println("Random number (0 to 1): " + Math.random());
  System.out.println("Rounded value of 54.42: " + Math.round(54.42));
  System.out.println("Ceil of 42.7: " + Math.ceil(42.7));
  System.out.println("Floor of 42.7: " + Math.floor(42.7));
 }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Packages>javac Packages2.java

C:\Rahi Pics\Java Programs\Packages>java Packages2
Square root of 9.0: 3.0
Absolute value of -10.5: 10.5
Power (3^5): 243.0
Maximum of 100 and 120: 120
Random number (0 to 1): 0.5160827038860766
Rounded value of 54.42: 54
Ceil of 42.7: 43.0
Floor of 42.7: 42.0
```

User defined Packages:

3.Packages3:

Package:

```
package school.students;

public class Student{

 public void display(){

    System.out.println("Student class from school.students package");

 }

}
```

Code:

```
import school.students.Student;

public class Packages3{

 public static void main(String[] args){

  Student s = new Student();

  s.display();

 }

}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Packages>javac -d . Student.java

C:\Rahi Pics\Java Programs\Packages>javac Packages3.java

C:\Rahi Pics\Java Programs\Packages>java Packages3
Student class from school.students package
```

4.Packages4:

Package:

```
package vehicles;
public class Vehicle{
  public void ride(){
    System.out.println("Riding a Duke.");
  }
}
```

Code:

```
import vehicles.Vehicle;
public class Packages4{
  public static void main(String[] args){
    Vehicle bike = new Vehicle();
    bike.ride();
  }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Packages>javac -d . Vehicle.java

C:\Rahi Pics\Java Programs\Packages>javac Packages4.java

C:\Rahi Pics\Java Programs\Packages>java Packages4
Riding a Duke.
```

File Handling:

1.File1:

```
import java.io.FileWriter;
import java.io.IOException;
public class File1{
   public static void main(String[] args) {
      try {
         FileWriter writer = new FileWriter("example.txt");
         writer.write("Hello, this is a file handling example in Java!");
         writer.close();
         System.out.println("File written successfully.");
      } catch (IOException e) {
         System.out.println("An error occurred.");
         e.printStackTrace();
      }
   }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\File Handling>javac File1.java

C:\Rahi Pics\Java Programs\File Handling>java File1
File written successfully.
```

2.File2:

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
public class File2 {
   public static void main(String[] args) {
      try {
         File file = new File("example.txt");
         Scanner reader = new Scanner(file);
         while (reader.hasNextLine()) {
            String data = reader.nextLine();
            System.out.println(data);
         }
         reader.close();
      } catch (FileNotFoundException e) {
         System.out.println("File not found.");
         e.printStackTrace();
      }
   }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\File Handling>javac File2.java

C:\Rahi Pics\Java Programs\File Handling>java File2
Hello, this is a file handling example in Java!
```

3.File3:

```
import java.io.FileWriter;
import java.io.IOException;
public class File3 {
   public static void main(String[] args) {
     try {
        FileWriter writer = new FileWriter("example.txt", true); // true = append mode
        writer.write("\nAppended line using FileWriter.");
        writer.close();
        System.out.println("Data appended successfully.");
      } catch (IOException e) {
        System.out.println("Error appending to file.");
        e.printStackTrace();
      }
   }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\File Handling>javac File3.java

C:\Rahi Pics\Java Programs\File Handling>java File3
Data appended successfully.
```

4.File4:

```
import java.io.File;
public class File4 {
   public static void main(String[] args) {
      File file = new File("example.txt");
      if (file.delete()) {
         System.out.println("Deleted the file: " + file.getName());
      } else {
         System.out.println("Failed to delete the file.");
      }
   }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\File Handling>javac File4.java

C:\Rahi Pics\Java Programs\File Handling>java File4
Deleted the file: example.txt
```

Exceptional Handling:

1.Exceptional1:

```
public class Exceptional1{
  public static void main(String[] args) {
    try {
       String s = null;
          System.out.println(s.length());
       } catch (NullPointerException e) {
          System.out.println("Caught null pointer exception.");
       } finally {
          System.out.println("Finally block always runs.");
       }
    }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Exceptional Handling>javac Exceptional1.java

C:\Rahi Pics\Java Programs\Exceptional Handling>java Exceptional1
Caught null pointer exception.
Finally block always runs.
```

2.Exceptional2:

```
public class Exceptional2 {
   public static void main(String[] args) {
      try {
         int[] arr = new int[5];
         arr[10] = 50; // ArrayIndexOutOfBoundsException
      } catch (ArithmeticException e) {
         System.out.println("Arithmetic Exception caught.");
      } catch (ArrayIndexOutOfBoundsException e) {
         System.out.println("Array index out of bounds!");
      } catch (Exception e) {
         System.out.println("Some other exception occurred.");
      }
   }
}
```

OUTPUT:

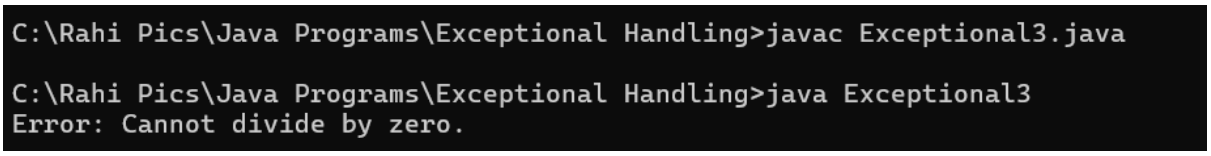
```
C:\Rahi Pics\Java Programs\Exceptional Handling>javac Exceptional2.java

C:\Rahi Pics\Java Programs\Exceptional Handling>java Exceptional2
Array index out of bounds!
```

3.Exceptional3:

```
public class Exceptional3{
   public static void main(String[] args) {
      try {
         int a = 10 / 0; // ArithmeticException
      } catch (ArithmeticException e) {
         System.out.println("Error: Cannot divide by zero.");
      }
   }
}
```

OUTPUT:



4.Exceptional4:

```
class AgeException extends Exception {
   AgeException(String message) {
      super(message);
   }
}
public class Exceptional4 {
   public static void checkAge(int age) throws AgeException {
      if (age < 18)
         throw new AgeException("Age must be 18 or above.");
      else
         System.out.println("Age is valid.");
   }
   public static void main(String[] args) {
      try {
         checkAge(16);
      } catch (AgeException e) {
         System.out.println("Exception: " + e.getMessage());
      }
   }
}
```

OUTPUT:

```
C:\Rahi Pics\Java Programs\Exceptional Handling>javac Exceptional4.java

C:\Rahi Pics\Java Programs\Exceptional Handling>java Exceptional4
Exception: Age must be 18 or above.
```