
# LAB RECORD

23CSE111- Object Oriented Programming

***Submitted by***

CH.SC.U4CSE24149 -Vighranth Sk

**BACHELOR OF TECHNOLOGY**

IN

COMPUTER SCIENCE AND ENGINEERING

AMRITA

VISHWA VIDYAPEETHAM

AMRITA

SCHOOL OF COMPUTING

CHENNAI

March - 2025

**AMRITA VISHWA**

**VIDYAPEETHAM AMRITA SCHOOL**

**OF COMPUTING, CHENNAI**

# BONAFIDE CERTIFICATE

This is to certify that the Lab Record work for 23CSE111- Object Oriented Programming Subject submitted by ***CH.SC.U4CSE24149 – Vighranth SK*** in **"Computer Science and Engineering"** is a Bonafide record of the work carried out under my guidance and supervision at Amrita School of Computing, Chennai.

This Lab examination held on / /2025

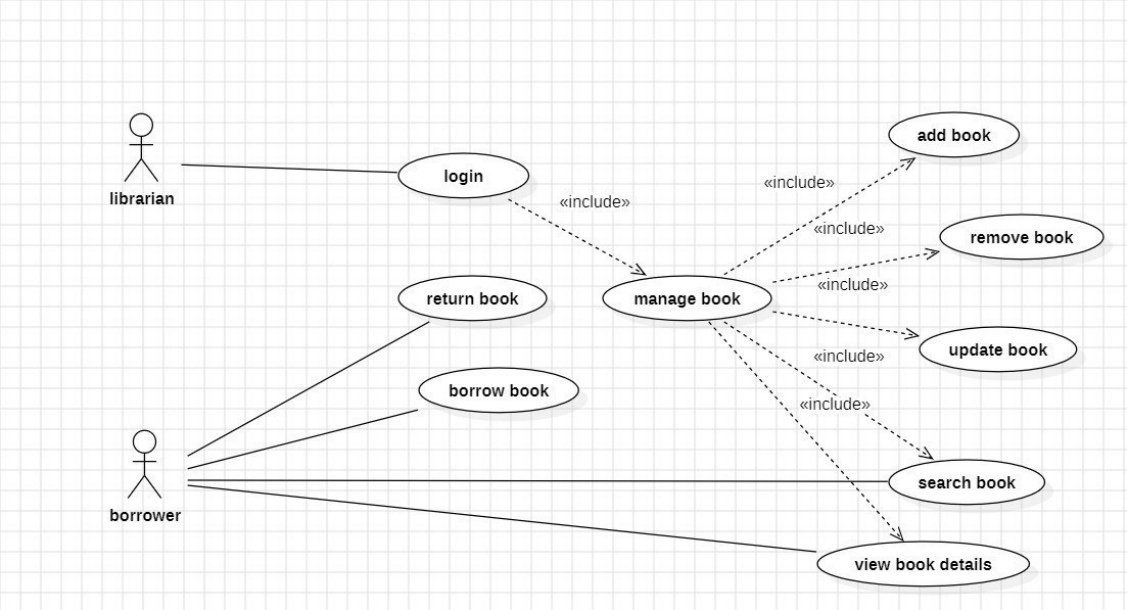
Internal Examiner 1 Internal Examiner 2

# INDEX

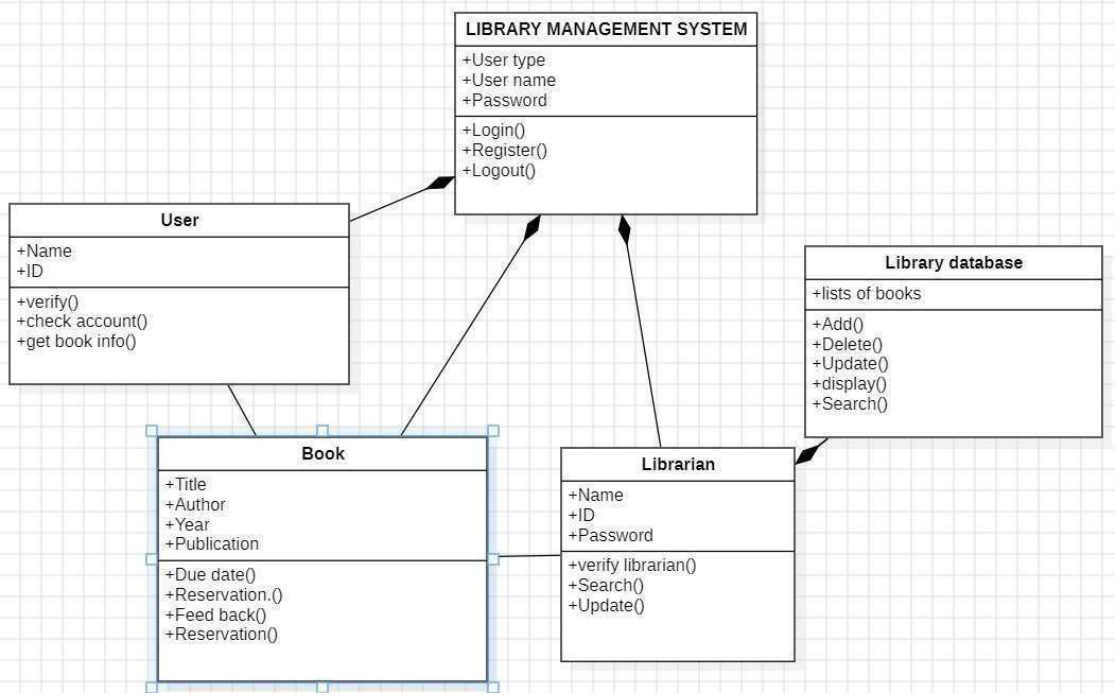
# UML DIAGRAMS

1. LIBRARY MANAGEMENT

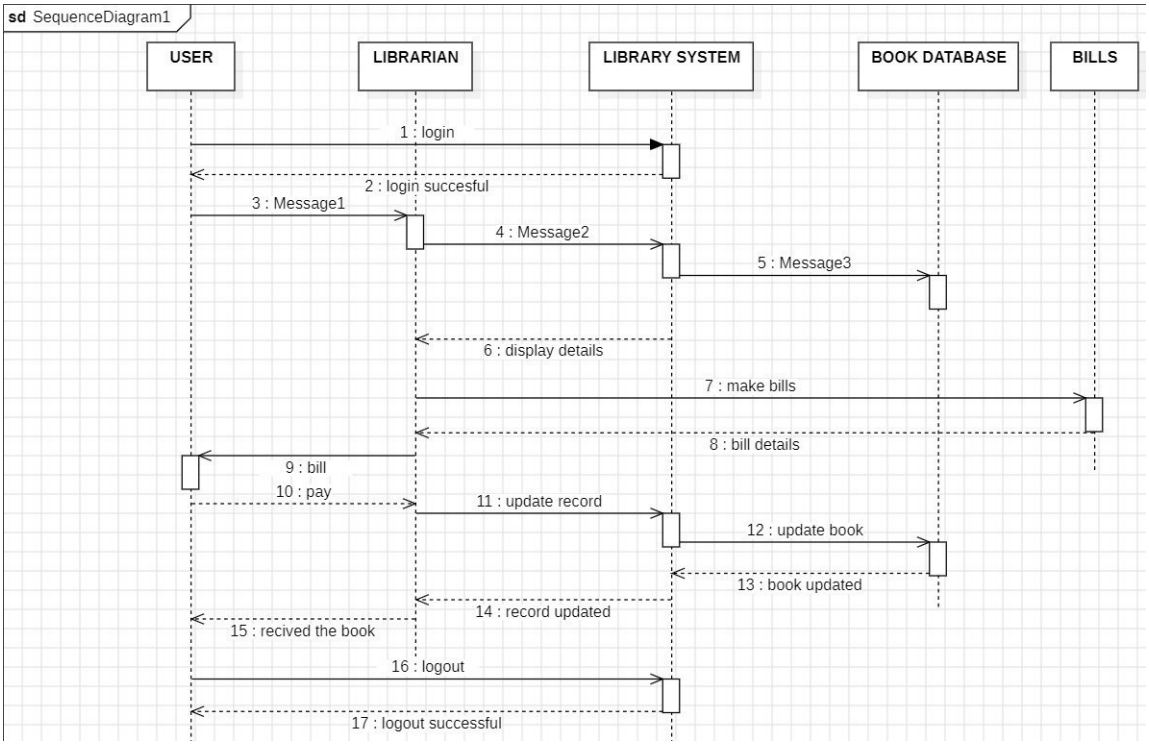
a) USE CASE DIAGRAM



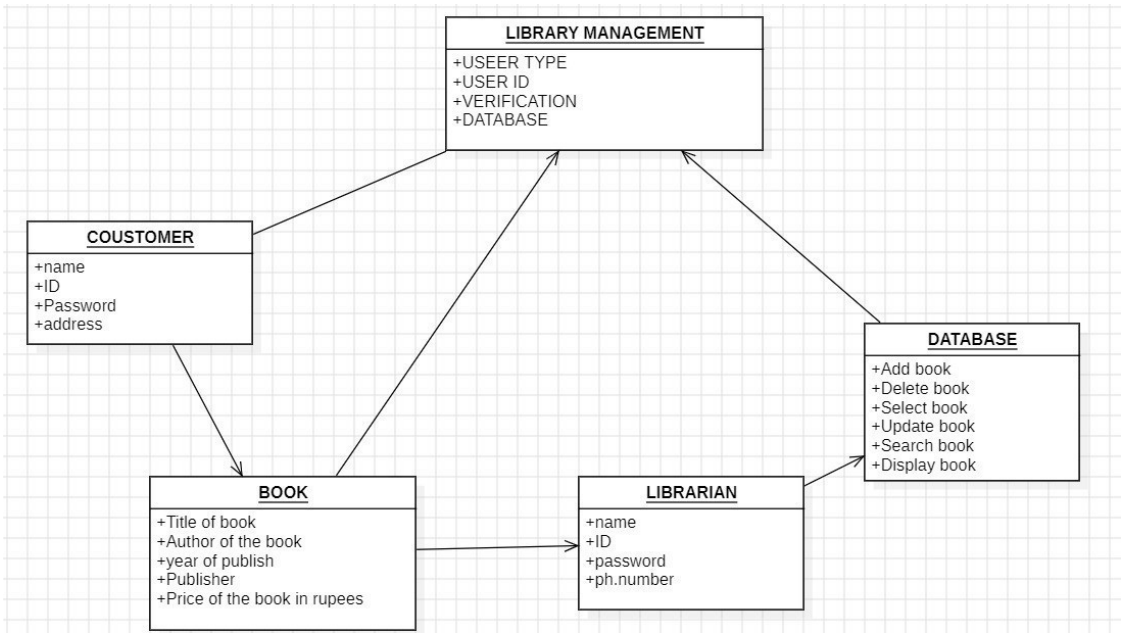

b) CLASS DIAGRAM



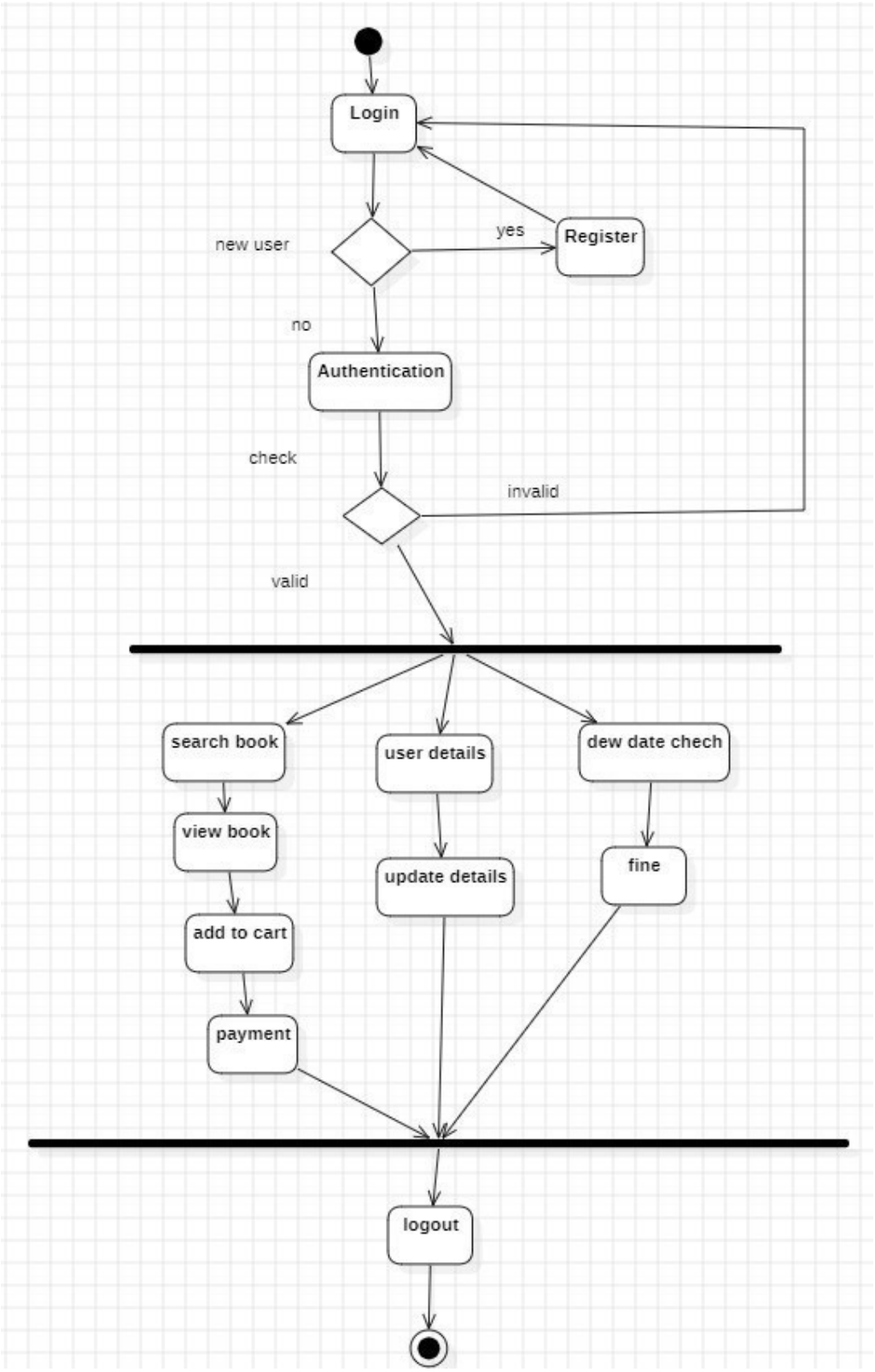
C) SEQUENCE DIAGRAM




d) OBJECT DIAGRAM

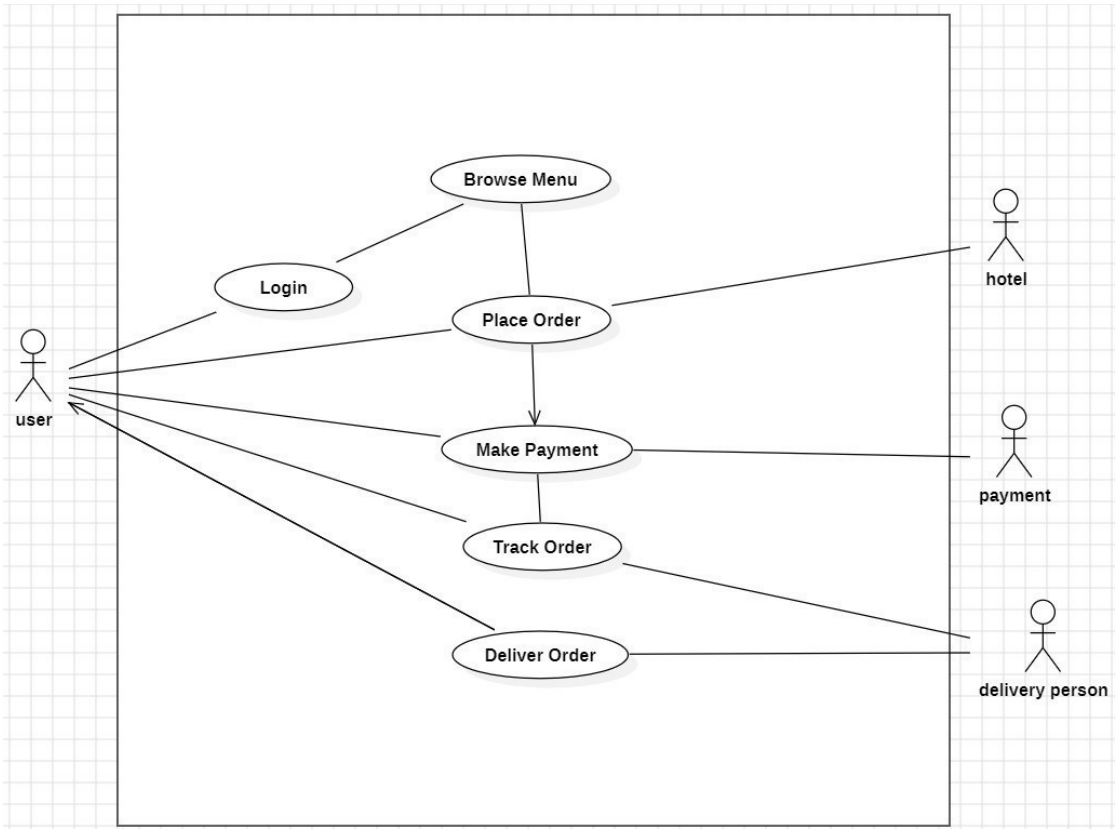


e) STATE-ACTIVITY DIAGRAM

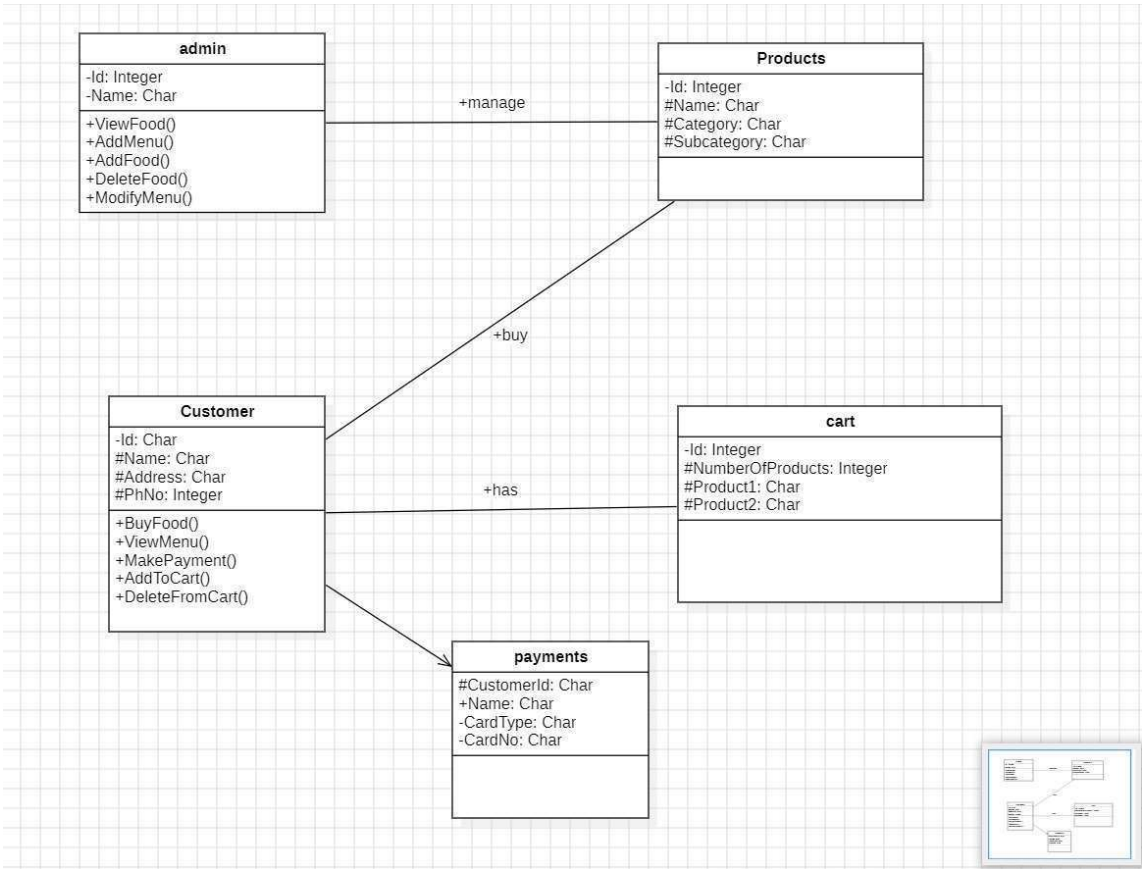


# 2. FOOD ORDERING MANAGEMENT

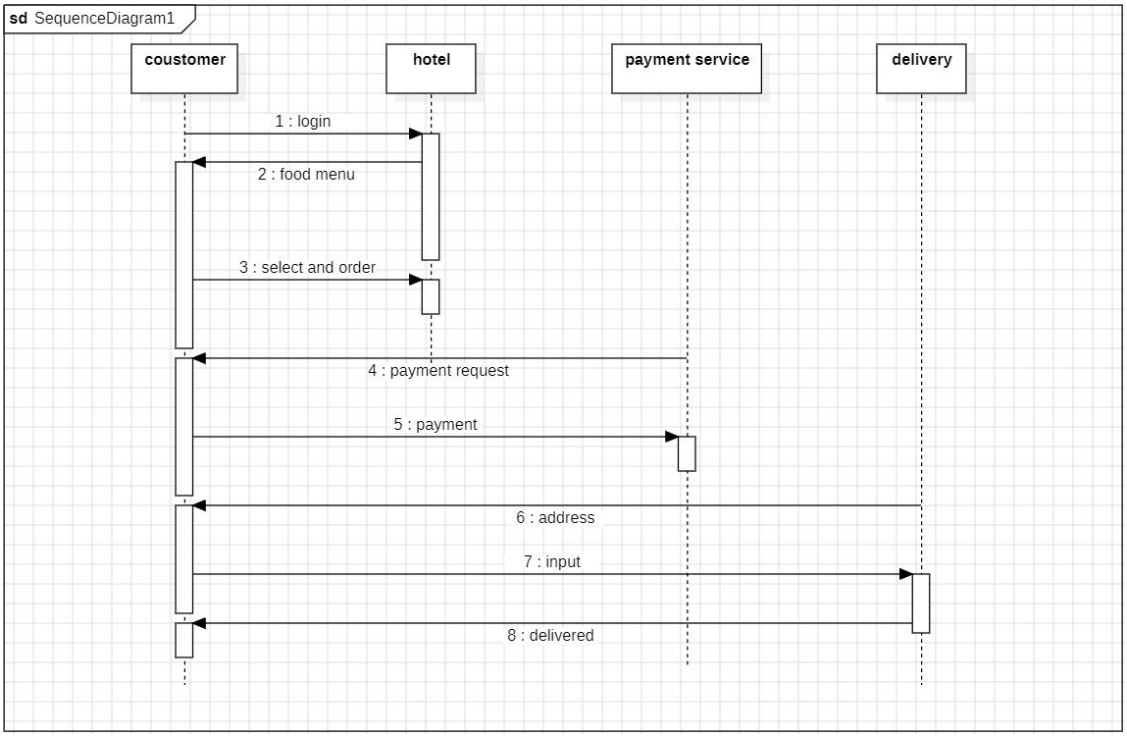
a) USE CASE DIAGRAM



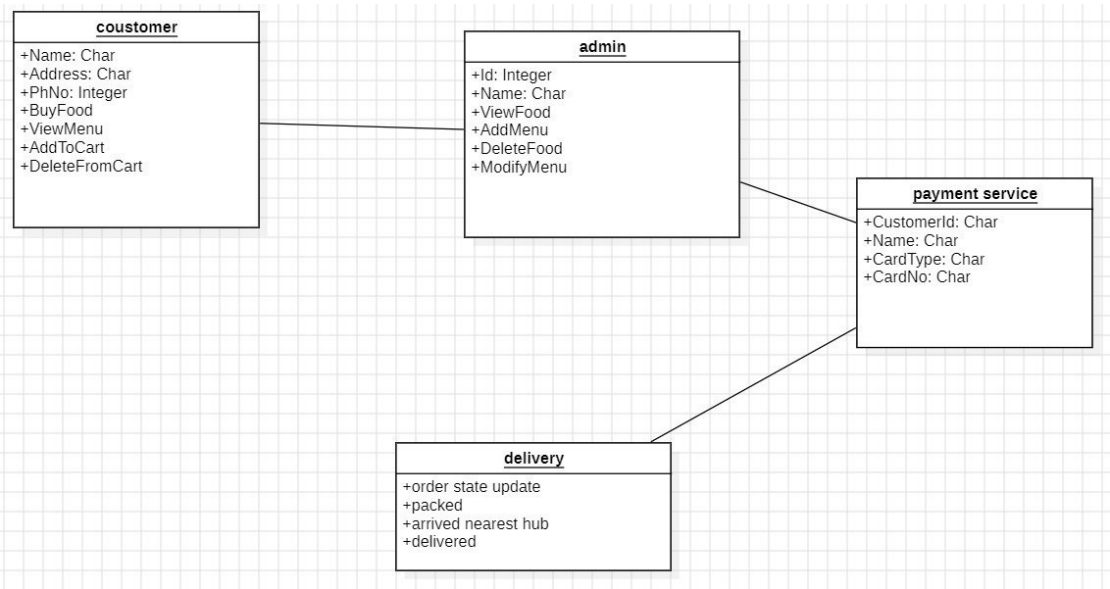
b) CLASS DIAGRAM



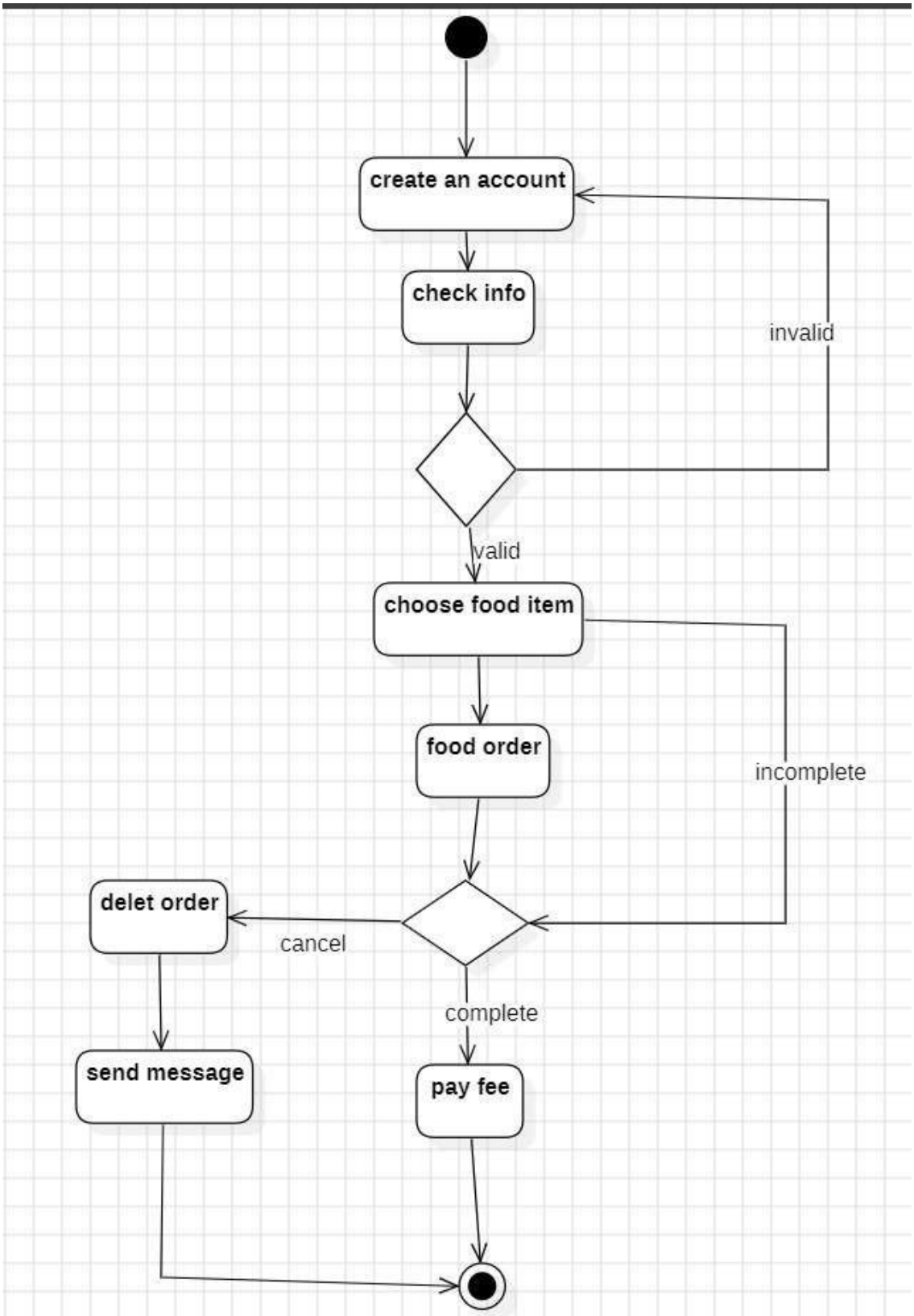

c) SEQUENCE DIAGRAM



d) OBJECT DIAGRAM



e) STATE-ACTIVITY DIAGRAM



# 3. BASIC JAVA PROGRAMS

a) Simple Addition

code:

```
public class Addition {
   public static void main(String[] args) {
      int num1 = 10, num2 = 20, sum;
      sum = num1 + num2;
      System.out.println("Sum: " + sum);
   }
}
```

output:

```
Sum: 30
```

b) Odd or Even

code:

```
import java.util.Scanner;


public class OddEven {
   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.");
      }
   }
```

output:

```
15 is odd.
```

c) Simple Calculator

code:

```
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.println("Select operation (+, -, *, /): ");
      char operator = sc.next().charAt(0);

      double       result;
      switch (operator) {
         case '+': result = num1 + num2; break;
         case '-': result = num1 - num2; break;
         case '*': result = num1 * num2; break;
         case '/': result = num1 / num2; break;
         default: System.out.println("Invalid operator!"); return;
      }
      System.out.println("Result: " + result);
   }
}
```

output:

```
Result: 50.0
```

d) 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();
      long fact = 1;
      for (int i = 1; i <= num; i++) {
         fact *= i;
      }
      System.out.println("Factorial of " + num + " is " + fact);
   }
}
```

output:

```
Factorial of 5 is 120
```

e) Prime Number Check

code:

```
import java.util.Scanner;

public class PrimeNumber {
   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;

      for (int i = 2; i <= num / 2; i++) {
```

```
            if (num % i == 0) {
                isPrime = false;
                break;
            }
        }

        if (isPrime && num > 1) {
            System.out.println(num + " is a prime number.");
        } else {
            System.out.println(num + " is not a prime number.");
        }
    }
}
```

output:

```
7 is a prime number.
```

f) Fibonacci Series

code:

```
import java.util.Scanner;

public class Fibonacci {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of terms: ");
        int terms = sc.nextInt();
        int n1 = 0, n2 = 1, n3;

        System.out.print(n1 + " " + n2 + " ");
        for (int i = 2; i < terms; i++) {
            n3 = n1 + n2;
            System.out.print(n3 + " ");
```

```
            n1 = n2;
            n2 = n3;
        }
    }
}
```

output:

```
0 1 1 2 3 5 8
```

g) Array Sum

code:

```
public class ArraySum {
    public static void main(String[] args) {
        int[] numbers = {10, 20, 30, 40, 50};
        int sum = 0;

        for (int num : numbers) {
            sum += num;
        }

        System.out.println("Sum of array elements: " + sum);
    }
}
```

output:

```
Sum of array elements: 150
```

h) Palindrome Check

code:

```
import java.util.Scanner;

public class Palindrome {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter a string: ");
      String str = sc.nextLine();
      String reversed = new StringBuilder(str).reverse().toString();

      if (str.equals(reversed)) {
         System.out.println(str + " is a palindrome.");
      } else {
         System.out.println(str + " is not a palindrome.");
      }
   }
}
```

output:

```
madam is a palindrome.
```

i) Multiplication Table

code:

```
import java.util.Scanner;

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

```
        System.out.println("Multiplication Table of " + num + ":");
        for (int i = 1; i <= 10; i++) {
            System.out.println(num + " x " + i + " = " + (num * i));
        }
    }
}
```

output:

```
Multiplication Table of 5:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

j) Reverse a Number

code:

```
import java.util.Scanner;

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

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

```
        }


        System.out.println("Reversed number: " + reversed);
    }
}
```

output:

```
Reversed number: 54321
```

# INHERITANCE

## 4. SINGLE INHERITANCE PROGRAMS

### 4.a)

### CODE:

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

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

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

```
C:\Users\YourName\Desktop>javac Main.java
C:\Users\YourName\Desktop>java Main
This animal eats food.
The dog barks.
```

4.b)

CODE:

```
class Vehicle {
    public void start() {
        System.out.println("The vehicle is starting.");
    }
}

class Car extends Vehicle {
    public void honk() {
        System.out.println("The car is honking.");
    }
}

public class Main {
    public static void main(String[] args) {
        Car car = new Car();
        car.start();
        car.honk();
    }
}
```

OUTPUT:

```
C:\Users\YourName\Desktop>javac Main.java
C:\Users\YourName\Desktop>java Main
The vehicle is starting.
The car is honking.
```

## 5. MULTILEVEL INHERITANCE PROGRAMS

**5.a)**

**CODE:**

```
class Animal {
    public void eat() {
        System.out.println("Animal is eating.");
    }
}

class Mammal extends Animal {
    public void sleep() {
```

```
        System.out.println("Mammal is sleeping.");
    }
}

class Dog extends Mammal {
    public void bark() {
        System.out.println("Dog is barking.");
    }
}

public class Main {
    public static void main(String[] args) {
        Dog myDog = new Dog();

        myDog.eat();
        myDog.sleep();
        myDog.bark();
    }
}
```

**OUTPUT:**

```
C:\Users\YourName\Desktop>javac Main.java
C:\Users\YourName\Desktop>java Main
Animal is eating.
Mammal is sleeping.
Dog is barking.
```

**5.b)**

**CODE:**

```
class Vehicle {
    public void start() {
        System.out.println("The vehicle is starting.");
    }
}

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

class ElectricCar extends Car {
    public void charge() {
        System.out.println("The electric car is charging.");
    }
}

public class Main {
    public static void main(String[] args) {
        ElectricCar myElectricCar = new ElectricCar();
        myElectricCar.start();
        myElectricCar.drive();
        myElectricCar.charge();
    }
}
```

**OUTPUT:**

```
C:\Users\YourName\Desktop>javac Main.java
C:\Users\YourName\Desktop>java Main
The vehicle is starting.
The car is driving.
The electric car is charging.
```

# 6. HIERARCHICAL INHERITANCE PROGRAMS

**6.a)**

**CODE:**

```
class Animal {
    public void eat() {
        System.out.println("Animal is eating.");
    }
}

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

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

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

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

**OUTPUT:**

```
C:\Users\YourName\Desktop>javac Main.java
C:\Users\YourName\Desktop>java Main
Animal is eating.
Dog is barking.
Animal is eating.
Cat is meowing.
```

**6.b)**

**CODE:**

```
class Vehicle {
    public void start() {
        System.out.println("Vehicle is starting.");
    }
}

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

public class Main {
    public static void main(String[] args) {
        Car car = new Car();
        car.start();
        car.drive();

        Bike bike = new Bike();
        bike.start();
        bike.ride();
    }
}
```

**OUTPUT:**

```
C:\Users\YourName\Desktop>javac Main.java
C:\Users\YourName\Desktop>java Main
Vehicle is starting.
Car is driving.
Vehicle is starting.
Bike is riding.
```

# 7. HYBRID INHERITANCE PROGRAMS

**7.a)**

**CODE:**

```
class Animal:
    def sound(self):
        print("Animal sound")

class Mammal(Animal):
    def breathe(self):
        print("Breathing oxygen")

class Bird(Animal):
    def fly(self):
        print("Flying in the sky")

class Bat(Mammal, Bird):
    def eat(self):
        print("Bat eats insects")

bat = Bat()

bat.sound()
bat.breathe()
bat.fly()
bat.eat()
```

**OUTPUT:**

```
C:\Users\YourName\Desktop>python main.py
Animal sound
Breathing oxygen
Flying in the sky
Bat eats insects
```

**7.b)**

**CODE:**

```
#include<iostream>
using namespace std;

class Animal {
public:
   void sound() {
      cout << "Animal sound" << endl;
   }
};

class Mammal : public Animal {
public:
   void breathe() {
      cout << "Breathing oxygen" << endl;
   }
};

class Bird : public Animal {
public:
   void fly() {
      cout << "Flying in the sky" << endl;
   }
};

class Bat : public Mammal, public Bird {
public:
   void eat() {
      cout << "Bat eats insects" << endl;
   }
};

int main() {
   Bat bat;
   bat.sound();
   bat.breathe();
   bat.fly();
   bat.eat();
   return 0;
}
```

**OUTPUT:**

```
C:\Users\YourName\Desktop>g++ main.cpp -o main
C:\Users\YourName\Desktop>main
Animal sound
Breathing oxygen
Flying in the sky
Bat eats insects
```

# POLYMORPHISM

## 8. CONSTRUCTOR PROGRAMS

**8.a)**

**CODE:**

```
class Vehicle {
   String type;
   int wheels;

   Vehicle() {
      this.type = "Unknown";
      this.wheels = 0;
   }

   Vehicle(String type) {
      this.type = type;
      this.wheels = 4;
   }

   Vehicle(String type, int wheels) {
      this.type = type;
      this.wheels = wheels;
   }

   void display() {
       System.out.println("Vehicle Type: " + type + ", Wheels: " + wheels);
   }

   public static void main(String[] args) {
      Vehicle v1 = new Vehicle();
```

```
        Vehicle v2 = new Vehicle("Car");
        Vehicle v3 = new Vehicle("Bike", 2);

        v1.display();
        v2.display();
        v3.display();
    }
}
```

**OUTPUT:**

```
C:\Users\YourName\Desktop>javac Vehicle.java
C:\Users\YourName\Desktop>java Vehicle
Vehicle Type: Unknown, Wheels: 0
Vehicle Type: Car, Wheels: 4
Vehicle Type: Bike, Wheels: 2
```

# CONSTRUCTOR OVERLOADING

**9.a)**

**CODE;**

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

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

    Person(String name) {
        this.name = name;
        this.age = 18;
    }

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

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

```
    public static void main(String[] args) {
        Person p1 = new Person();
        Person p2 = new Person("Alice");
        Person p3 = new Person("Bob", 25);

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

**OUTPUT:**

```
C:\Users\YourName\Desktop>javac Person.java
C:\Users\YourName\Desktop>java Person
Name: Unknown, Age: 0
Name: Alice, Age: 18
Name: Bob, Age: 25
```

# 10.METHOD OVERLOADING PROGRAMS

10.a)
CODE:

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

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

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

   public static void main(String[] args) {
      Calculator calc = new Calculator();

      System.out.println("Sum of 2 integers: " + calc.add(5, 10));
      System.out.println("Sum of 2 doubles: " + calc.add(5.5, 2.3));
      System.out.println("Sum of 3 integers: " + calc.add(2, 3, 4));
   }
}
```

OUTPUT:

```
C:\Users\YourName\Desktop>javac Calculator.java
C:\Users\YourName\Desktop>java Calculator
Sum of 2 integers: 15
Sum of 2 doubles: 7.8
Sum of 3 integers: 9
```

10.b)
CODE:

```
class AreaCalculator {
   double area(double radius) {
      return 3.14 * radius * radius;
   }

   int area(int side) {
      return side * side;
   }

   int area(int length, int width) {
      return length * width;
   }

   public static void main(String[] args) {
      AreaCalculator obj = new AreaCalculator();

      System.out.println("Area of Circle: " + obj.area(5.5));
      System.out.println("Area of Square: " + obj.area(4));
      System.out.println("Area of Rectangle: " + obj.area(4, 6));
   }
}
```

OUTPUT:

```
C:\Users\Vighranth\Desktop>javac AreaCalculator.java
C:\Users\Vighranth\Desktop>java AreaCalculator
Area of Circle: 94.985
Area of Square: 16
Area of Rectangle: 24
```

# 11.METHOD OVERRIDING PROGRAMS

**11.a)**
**CODE:**

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

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

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

public class Main {
    public static void main(String[] args) {
        Animal a1 = new Dog();
        Animal a2 = new Cat();

        a1.makeSound();
        a2.makeSound();
    }
}
```

**OUTPUT:**

```
C:\Users\Vighranth\Desktop>javac Main.java
C:\Users\Vighranth\Desktop>java Main
Dog barks.
Cat meows.
```

**11.b)**

**CODE:**

```
class Bank {
    double getInterestRate() {
        return 0.0;
    }
}

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

class HDFC extends Bank {
    double getInterestRate() {
        return 6.8;
    }
}

public class Main {
    public static void main(String[] args) {
        Bank b1 = new SBI();
        Bank b2 = new HDFC();

        System.out.println("SBI Interest Rate: " + b1.getInterestRate() + "%");
        System.out.println("HDFC Interest Rate: " + b2.getInterestRate() + "%");
    }
}
```

**OUTPUT:**

```
C:\Users\Vighranth\Desktop>javac Main.java
C:\Users\Vighranth\Desktop>java Main
SBI Interest Rate: 5.5%
HDFC Interest Rate: 6.8%
```

# ABSTRACTION

## 12.INTERFACE PROGRAMS

**12.a)**

**CODE:**

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

   void display() {
      System.out.println("Vehicle is ready to use.");
   }
}

class Car extends Vehicle {
   void start() {
      System.out.println("Car starts with a key.");
   }
}

class Bike extends Vehicle {
   void start() {
      System.out.println("Bike starts with a self-start button.");
   }
}

public class Main {
   public static void main(String[] args) {
      Vehicle v1 = new Car();
      Vehicle v2 = new Bike();

      v1.start();
      v2.start();
      v1.display();
   }
}
```

**OUTPUT:**

```
C:\Users\Vighranth\Desktop>javac Main.java
C:\Users\Vighranth\Desktop>java Main
Car starts with a key.
Bike starts with a self-start button.
Vehicle is ready to use.
```

**12.b)**

**CODE:**

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

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

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

public class Main {
   public static void main(String[] args) {
      Animal a1 = new Dog();
      Animal a2 = new Cat();

      a1.makeSound();
      a2.makeSound();
   }
}
```

**OUTPUT:**

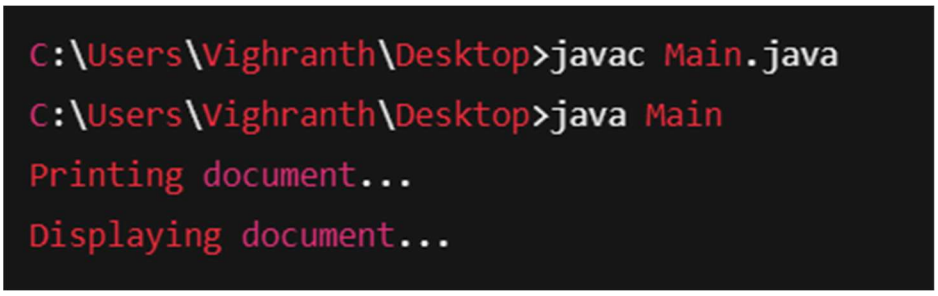
```
C:\Users\Vighranth\Desktop>javac Main.java
C:\Users\Vighranth\Desktop>java Main
Dog barks.
Cat meows.
```

**12.c)**

**CODE:**

```
interface Printable {
    void print();
}

interface Showable {
    void show();
}

class Document implements Printable, Showable {
    public void print() {
        System.out.println("Printing document...");


    }

    public void show() {
        System.out.println("Displaying document...");
    }
}

public class Main {
    public static void main(String[] args) {
        Document doc = new Document();
        doc.print();
        doc.show();
    }
}
```

**OUTPUT:**



**12.d)**

**CODE:**

```
interface Vehicle {
    void speedUp();

    default void applyBrakes() {
        System.out.println("Brakes applied.");
    }

    static void showMessage() {
        System.out.println("Vehicle interface in action.");
    }
}

class Car implements Vehicle {
    public void speedUp() {
        System.out.println("Car speeds up.");
    }
}

public class Main {
    public static void main(String[] args) {
        Car c = new Car();
        c.speedUp();
        c.applyBrakes();
        Vehicle.showMessage();
    }
}
```

**OUTPUT:**

```
C:\Users\Vighranth\Desktop>javac Main.java
C:\Users\Vighranth\Desktop>java Main
Car speeds up.
Brakes applied.
Vehicle interface in action.
```

# 13.ABSTRACT CLASS PROGRAMS

**13.a)**
**CODE:**

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

    void sleep() {
        System.out.println("Animal is sleeping...");
    }
}

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

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

public class Main {
    public static void main(String[] args) {
        Animal a1 = new Dog();
        Animal a2 = new Cat();

        a1.makeSound();
        a2.makeSound();
        a1.sleep();
    }
}
```

**OUTPUT:**

```
C:\Users\Vighranth\Desktop>javac Main.java
C:\Users\Vighranth\Desktop>java Main
Dog barks.
Cat meows.
Animal is sleeping...
```

**13.b)**
**CODE:**

```
abstract class Bank {
    abstract double getInterestRate();

    void display() {
        System.out.println("Banking system running...");
    }
}

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

class HDFC extends Bank {
    double getInterestRate() {
        return 6.8;
    }
}

public class Main {
    public static void main(String[] args) {
        Bank b1 = new SBI();
        Bank b2 = new HDFC();

        System.out.println("SBI Interest Rate: " + b1.getInterestRate() + "%");
        System.out.println("HDFC Interest Rate: " + b2.getInterestRate() + "%");
        b1.display();
    }
}
```

**OUTPUT:**

```
C:\Users\Vighranth\Desktop>javac Main.java
C:\Users\Vighranth\Desktop>java Main
SBI Interest Rate: 5.5%
HDFC Interest Rate: 6.8%
Banking system running...
```

**13.c)**

**CODE:**

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

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

    abstract double calculateSalary();

    void display() {
        System.out.println("Employee Name: " + name + ", ID: " + id);
    }
}

class FullTimeEmployee extends Employee {
    double salary;

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

    double calculateSalary() {
        return salary;
    }
}

class PartTimeEmployee extends Employee {
    double hourlyRate;
    int hoursWorked;

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

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

public class Main {
    public static void main(String[] args) {
        Employee e1 = new FullTimeEmployee("Alice", 101, 50000);
        Employee e2 = new PartTimeEmployee("Bob", 102, 20, 100);
```

```
    e1.display();
        System.out.println("Salary: $" + e1.calculateSalary());


        e2.display();
        System.out.println("Salary: $" + e2.calculateSalary());
    }
}
```

**OUTPUT:**

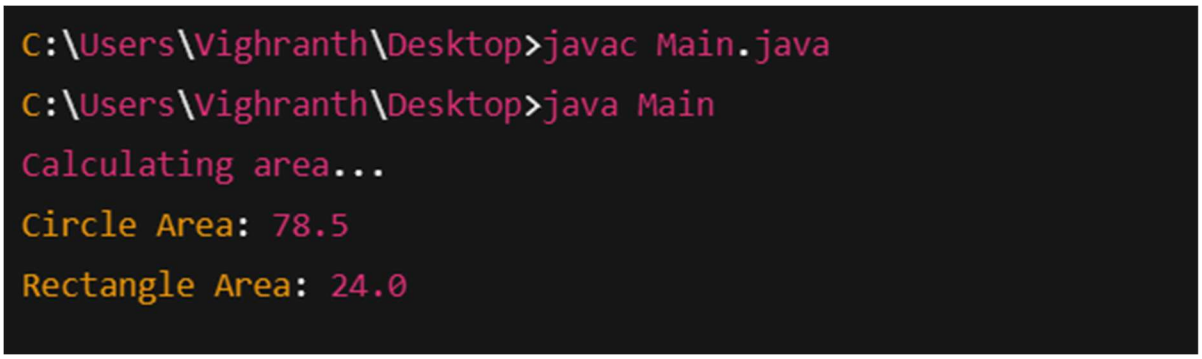
```
C:\Users\Vighranth\Desktop>javac Main.java
C:\Users\Vighranth\Desktop>java Main
Employee Name: Alice, ID: 101
Salary: $50000.0
Employee Name: Bob, ID: 102
Salary: $2000.0
```

**13.d)**
**CODE:**

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

    void display() {
        System.out.println("Calculating area...");
    }
}

class Circle extends Shape {
    double radius;

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

    double calculateArea() {
        return 3.14 * radius * radius;
    }
}

class Rectangle extends Shape {
    double length, width;

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

```
    }

    double calculateArea() {
        return length * width;
    }
}


public class Main {
    public static void main(String[] args) {
        Shape s1 = new Circle(5);
        Shape s2 = new Rectangle(4, 6);



 s1.display();


        System.out.println("Circle Area: " + s1.calculateArea());
        System.out.println("Rectangle Area: " + s2.calculateArea());
    }
}
```

**OUTPUT:**



# ENCAPSULATION

## 14.ENCAPSULATION PROGRAMS

14.a)

CODE:

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

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

   public void setAge(int age) {
      if (age > 0) {
         this.age = age;
      } else {
         System.out.println("Invalid  age");
      }
   }

   public String getName() {
      return name;
   }


 public int getAge() {
      return age;
   }

   public static void main(String[] args) {
      Student s = new Student();


      s.setName("John");
      s.setAge(20);

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

OUTPUT:

```
C:\Users\Vighranth\Desktop>javac Student.java
C:\Users\Vighranth\Desktop>java Student

Student Name: John
Student Age: 20
```

14.b)

CODE:

```
class BankAccount {
   private double balance;

   public void deposit(double amount) {
      if (amount > 0) {
         balance += amount;
         System.out.println("Deposited: $" + amount);
      } else {
         System.out.println("Invalid deposit amount");
      }
   }

   public void withdraw(double amount) {
      if (amount > 0 && amount <= balance) {
         balance -= amount;
         System.out.println("Withdrawn: $" + amount);
      } else {
         System.out.println("Insufficient funds or invalid amount");
      }
   }

   public double getBalance() {
      return balance;
   }

   public static void main(String[] args) {
      BankAccount account = new BankAccount();
      account.deposit(1000);
      account.withdraw(500);


 System.out.println("Balance: $" + account.getBalance());
   }
}
```

OUTPUT:

```
C:\Users\Vighranth\Desktop>javac BankAccount.java
C:\Users\Vighranth\Desktop>java BankAccount

Deposited: $1000.0
Withdrawn: $500.0
Balance: $500.0
```

14.c)

CODE:

```
class Employee {
   private String name;
   private double salary;

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

   public void setSalary(double salary) {
      if (salary > 0) {
         this.salary = salary;
      } else {
         System.out.println("Invalid salary amount");
      }
   }

   public String getName() {
      return name;
   }

   public double getSalary() {
      return salary;
   }

   public static void main(String[] args) {
      Employee e = new Employee();
      e.setName("Alice");
      e.setSalary(50000);

      System.out.println("Employee Name: " + e.getName());
      System.out.println("Employee Salary: $" + e.getSalary());
   }
}
```

OUTPUT:

```
C:\Users\Vighranth\Desktop>javac Employee.java
C:\Users\Vighranth\Desktop>java Employee

Employee Name: Alice
Employee Salary: $50000.0
```

14.d) CODE:

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

    public void setModel(String model) {
        this.model = model;
    }

    public void setSpeed(int speed) {
        if (speed >= 0) {
            this.speed = speed;
        } else {
            System.out.println("Invalid speed value");
        }
    }

    public String getModel() {
        return model;
    }

    public int getSpeed() {
        0return speed;
    }

    public static void main(String[] args) {
        Car c = new Car();
        c.setModel("Tesla");
        c.setSpeed(120);

        System.out.println("Car Model: " + c.getModel());
        System.out.println("Car Speed: " + c.getSpeed() + " km/h");
    }
}
```

OUTPUT:

```
C:\Users\Vighranth\Desktop>javac Car.java
C:\Users\Vighranth\Desktop>java Car

Car Model: Tesla
Car Speed: 120 km/h
```

# 15.PACKAGES PROGRAMS

15.a)User Defined Packages

CODE:

```
package mypackage;

public class Message {
   public void showMessage() {
      System.out.println("Hello from mypackage!");
   }
}
import mypackage.Message;

public class Main {
   public static void main(String[] args) {
      Message msg = new Message();
      msg.showMessage();
   }
}
```

OUTPUT:

```
C:\Users\Vighranth\Desktop>javac -d . Message.java
C:\Users\Vighranth\Desktop>javac Main.java
C:\Users\Vighranth\Desktop>java Main

Hello from mypackage!
```

15.b)User Defined Packages

CODE:

```
package mathoperations;

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

   public int multiply(int a, int b) {
      return a * b;
   }
}
package mathoperations;

public class AdvancedCalculator {
   public double power(double base, double exponent) {
      return Math.pow(base, exponent);
   }
}

import mathoperations.Calculator;
import mathoperations.AdvancedCalculator;

public class Main {
   public static void main(String[] args) {
      Calculator calc = new Calculator();
      AdvancedCalculator advCalc = new AdvancedCalculator();

      System.out.println("Addition: " + calc.add(5, 10));
      System.out.println("Multiplication: " + calc.multiply(3, 4));
      System.out.println("Power: " + advCalc.power(2, 3));
   }
}
```

OUTPUT:

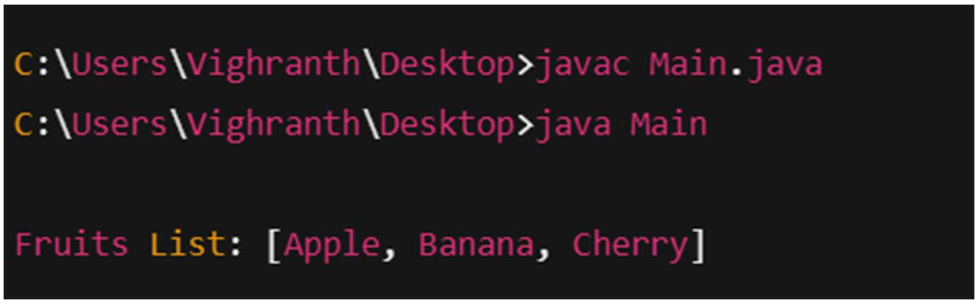
```
C:\Users\Vighranth\Desktop>javac -d . Calculator.java
C:\Users\Vighranth\Desktop>javac -d . AdvancedCalculator.java
C:\Users\Vighranth\Desktop>javac Main.java
C:\Users\Vighranth\Desktop>java Main

Addition: 15
Multiplication: 12
Power: 8.0
```

15.c)Built – in Package (3 Packages)

CODE:

```
import java.util.ArrayList;

public class Main {
   public static void main(String[] args) {
      ArrayList<String> list = new ArrayList<>();
      list.add("Apple");
      list.add("Banana");
      list.add("Cherry");

      System.out.println("Fruits List: " + list);
   }
}
```

OUTPUT:




15.d)Built – in Package (3 Packages)

CODE:

```
import java.io.File;
import java.io.IOException;

public class Main {
   public static void main(String[] args) {
      File file = new File("example.txt");
      try {
```

```
        if (file.createNewFile()) {
            System.out.println("File created: " + file.getName());
        } else {
            System.out.println("File already exists.");
        }
    } catch (IOException e) {
        System.out.println("An error occurred.");
    }
  }
}
```

OUTPUT:

```
C:\Users\Vighranth\Desktop>javac Main.java
C:\Users\Vighranth\Desktop>java Main

File created: example.txt
```

# 16.EXCEPTION HANDLING PROGRAMS

16.a)

CODE:

```
public class ExceptionExample1 {
  public static void main(String[] args) {
    try {
      int a = 10, b = 0;
      int result = a / b;
      System.out.println("Result: " + result);
    } catch (ArithmeticException e) {
      System.out.println("Error: Division by zero is not allowed.");
    }
  }
}
```

OUTPUT:

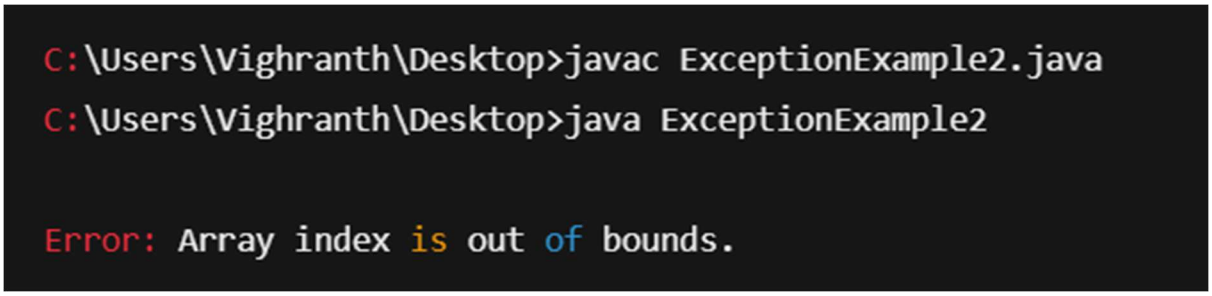
```
C:\Users\Vighranth\Desktop>javac ExceptionExample1.java
C:\Users\Vighranth\Desktop>java ExceptionExample1

Error: Division by zero is not allowed.
```

16.b)

C CODE:

```
public class ExceptionExample2 {
   public static void main(String[] args) {
      try {
         int[] arr = {1, 2, 3};
         System.out.println(arr[5]);
      } catch (ArrayIndexOutOfBoundsException e) {
         System.out.println("Error: Array index is out of bounds.");
      }

   }
}
```

OUTPUT:



16.c)

CODE:

```
public class ExceptionExample3 {
   public static void main(String[] args) {
      try {
          int num = Integer.parseInt("abc");
       } catch (NumberFormatException e) {
          System.out.println("Error: Cannot convert string to number.");
      } catch (Exception e) {
          System.out.println("General Exception Caught.");
      }
   }
}
```

OUTPUT:

```
C:\Users\Vighranth\Desktop>javac ExceptionExample3.java

C:\Users\Vighranth\Desktop>java ExceptionExample3
Error: Cannot convert string to number.
```

16.d)

CODE:

```
public class ExceptionExample4 {
   public static void main(String[] args) {
      try {
          int num = 5 / 0;
      } catch (ArithmeticException e) {
          System.out.println("Error: Division by zero.");
      } finally {
          System.out.println("Finally block executed.");
      }
   }
}
```

OUTPUT:

```
C:\Users\Vighranth\Desktop>javac ExceptionExample4.java

C:\Users\Vighranth\Desktop>java ExceptionExample4
Error: Division by zero.
Finally block executed.
```

# 17.FILE HANDLING PROGRAMS

17.a)

CODE:

```
import java.io.File;
import java.io.IOException;

public class CreateFileExample {
   public static void main(String[] args) {
      try {
         File file = new File("example.txt");
         if (file.createNewFile()) {
            System.out.println("File created: " + file.getName());
         } else {
            System.out.println("File already exists.");
         }
      } catch (IOException e) {
         System.out.println("An error occurred.");
      }
   }
}
```

OUTPUT:

```
C:\Users\Vighranth\Desktop>javac CreateFileExample.java

C:\Users\Vighranth\Desktop>java CreateFileExample
File created: example.txt

C:\Users\Vighranth\Desktop>
```

17.b)

CODE:

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

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

OUTPUT:

```
C:\Users\Vighranth\Desktop>javac WriteFileExample.java

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

17.c)

CODE:

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class ReadFileExample {
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("An error occurred.");
        }
    }
}
```

OUTPUT:

```
C:\Users\Vighranth\Desktop>javac ReadFileExample.java

C:\Users\Vighranth\Desktop>java ReadFileExample
Hello, this is a test file.
Welcome to Java file handling.
```

17.d)

CODE:

```
import java.io.File;

public class DeleteFileExample {
    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:\Users\Vighranth\Desktop>javac DeleteFileExample.java

C:\Users\Vighranth\Desktop>java DeleteFileExample
Deleted the file: example.txt
```