SCHOOL OF COMPUTING

R N VISHNU VIKAS

CH.SC.U4CSE24151

OBJECT ORIENTED PROGRAMMING (23CSE111)

LAB RECORD

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.U4CSE24151 – R N VISHNU VIKAS*** 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

Internal Examiner 1 Internal Examiner 2

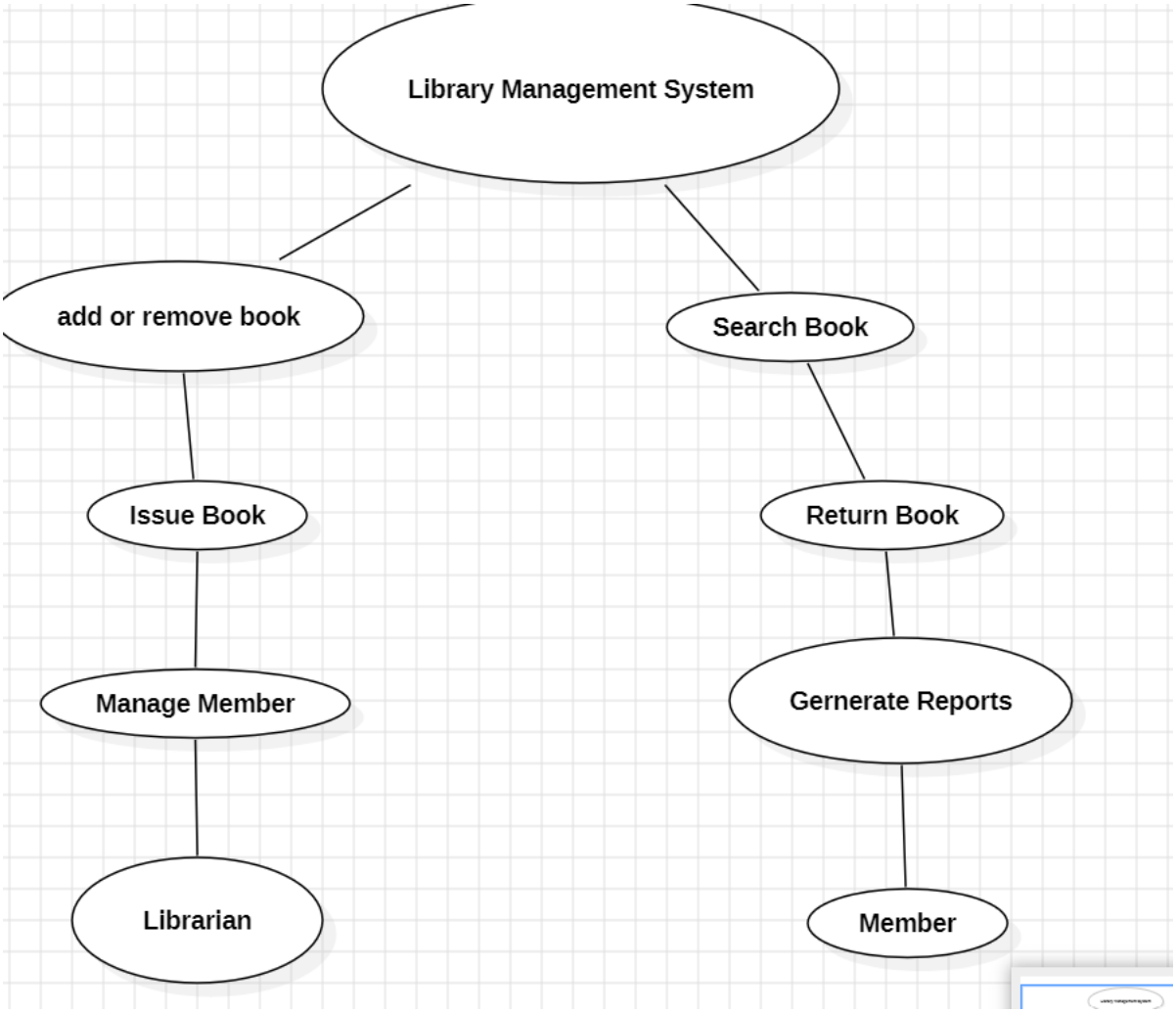
INDEX

	4.b)parent child	
5.	**MULTILEVEL INHERITANCE PROGRAM**	
	5.a)animal	
	5.b) Maruti 800	
6.	**HIERARCHICAL INHERITANCE PROGRAMS**	
	6.a)shape	
	6.b)bank	
7.	**HYBRID INHERITANCE PROGRAMS**	
	7.a) basic	
	7.b)grand parent	
	POLYMORPHISM	
8.	**CONSTRUCTOR PROGRAMS**	
	8.a)person	
9.	**CONSTRUCTOR OVERLOADING PROGRAMS**	
	9.a)student	
10.	**METHOD OVERLOADING PROGRAMS**	
	10.a)bank	
	10.b)argument	
11.	**METHOD OVERRIDING PROGRAMS**	
	11.a)animal	
	11.b)rbi	
	ABSTRACTION	
12.	**INTERFACE PROGRAMS**	
	12.a)shape	
	12.b)sports	
	12.c)employee	
	12.d)notification	
13.	**ABSTRACT CLASS PROGRAMS**	
	13.a)student teacher	
	13.b)login managment	
	13.c)addition subtraction	

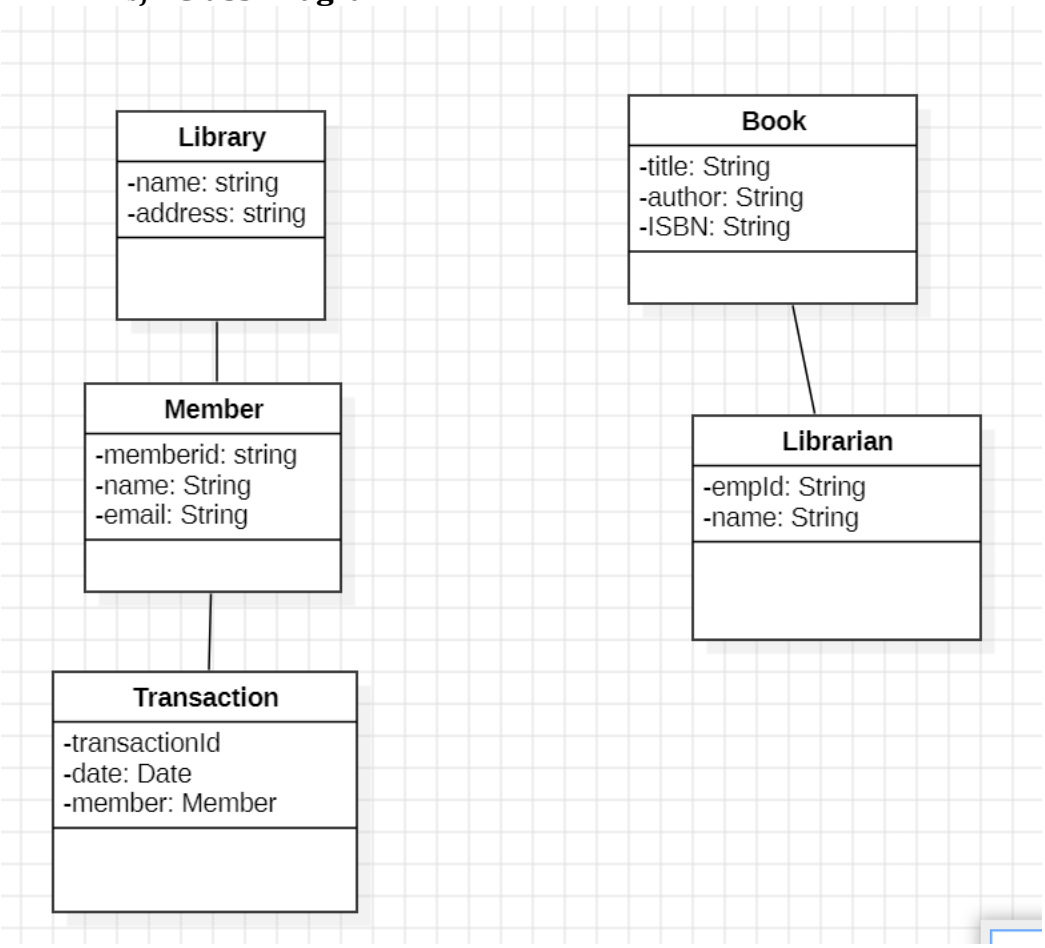
UML DIAGRAMS

1. LIBRARY MANAGEMENT SYSTEM

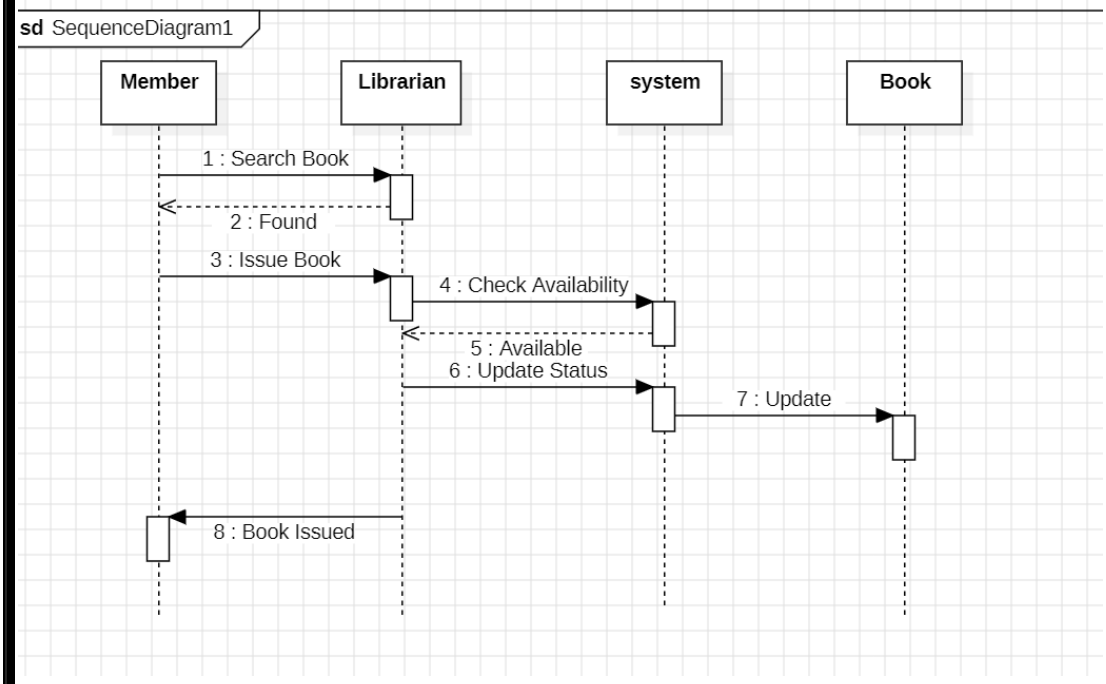
1.a) Use Case Diagram:



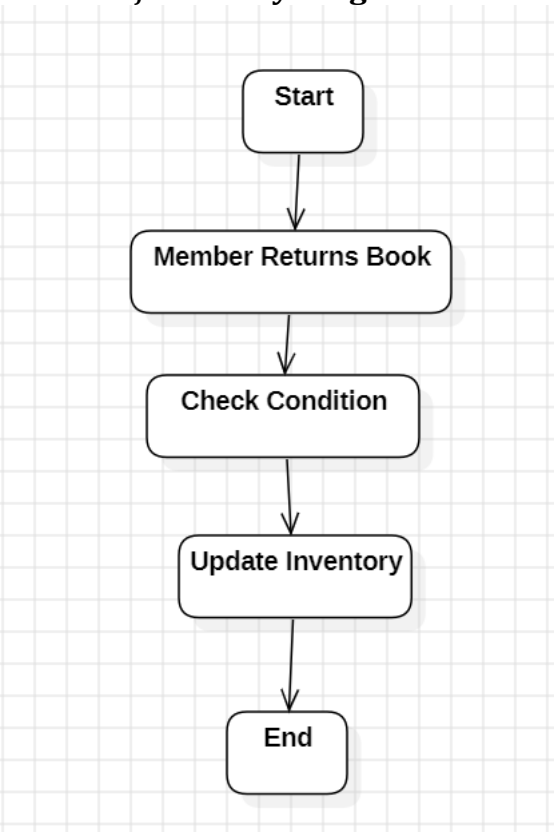
1.b) Class Diagram:



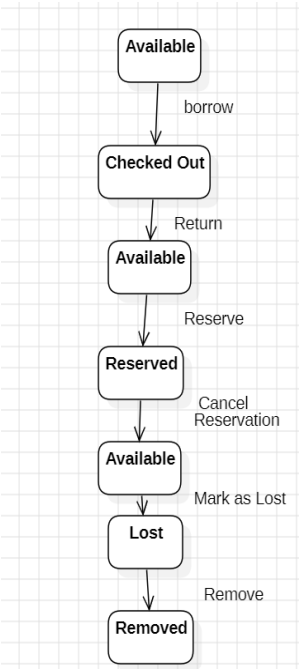
1.c) Sequence Diagram:



1.d) Activity Diagram:

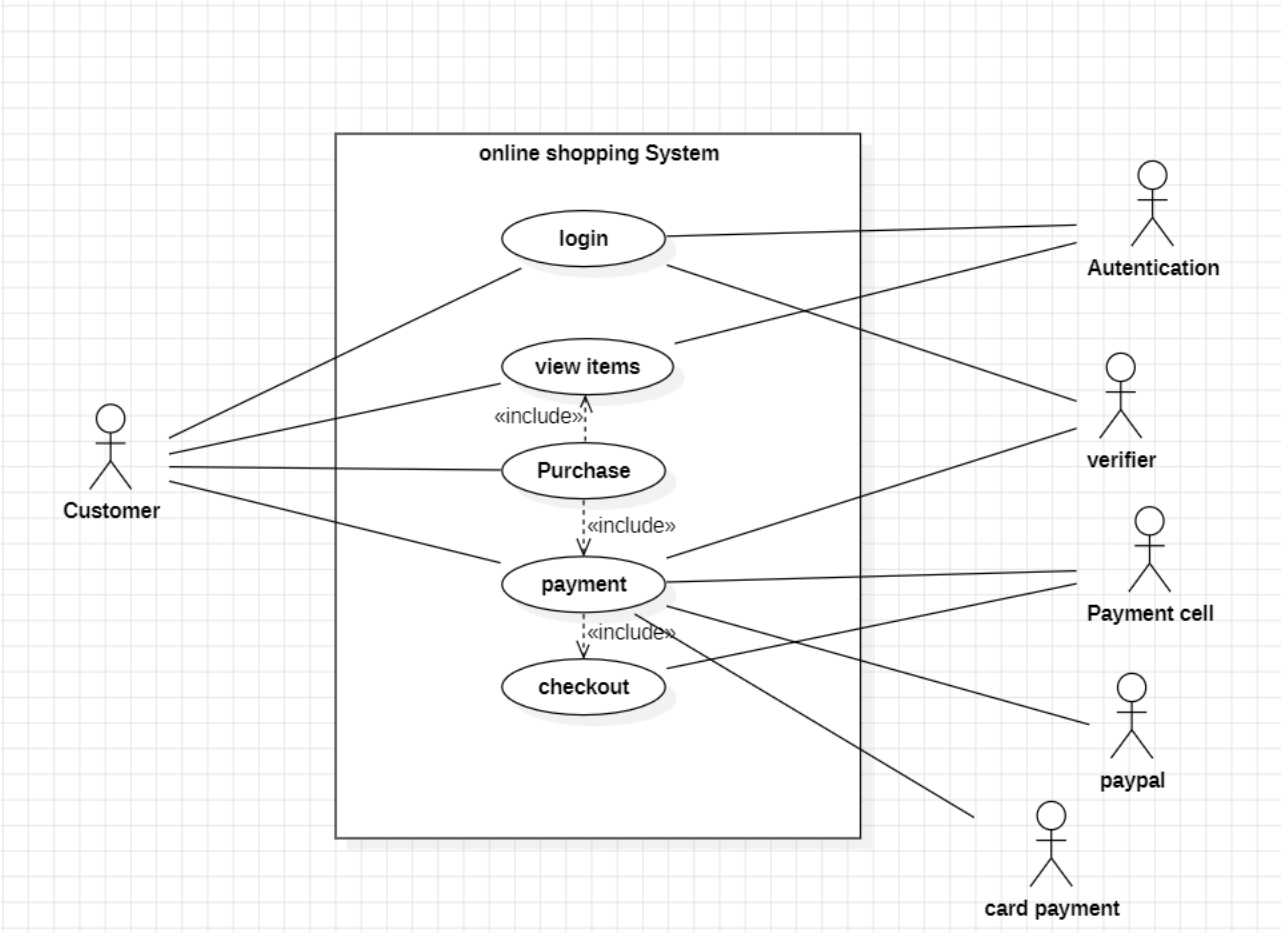


1.e) State-Activity Diagram:

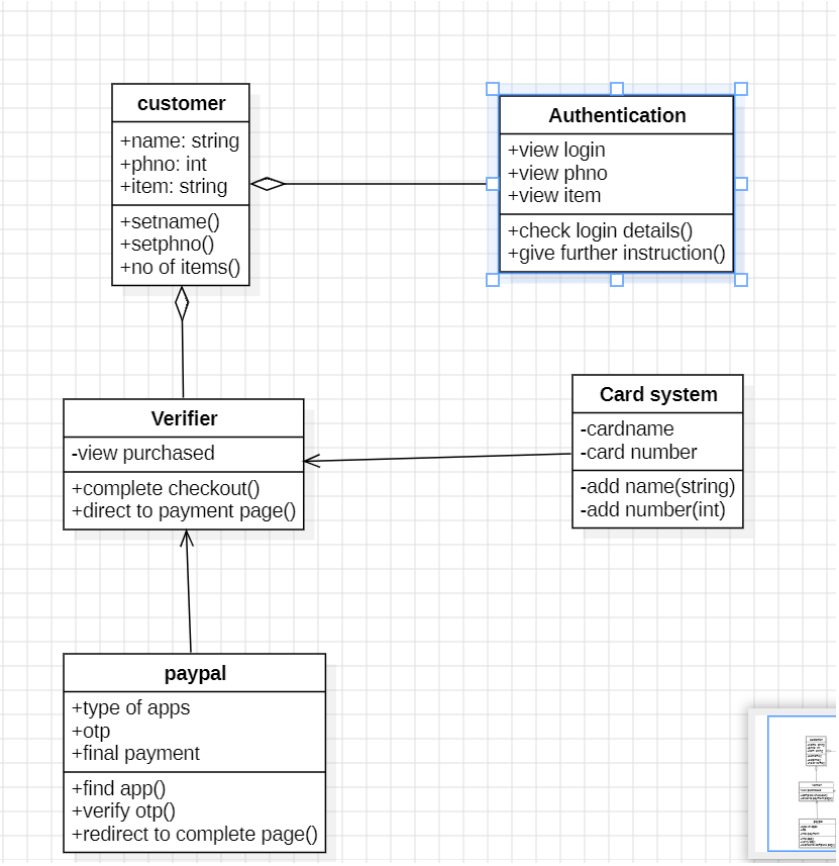


2. SHIPPING MANAGEMENT

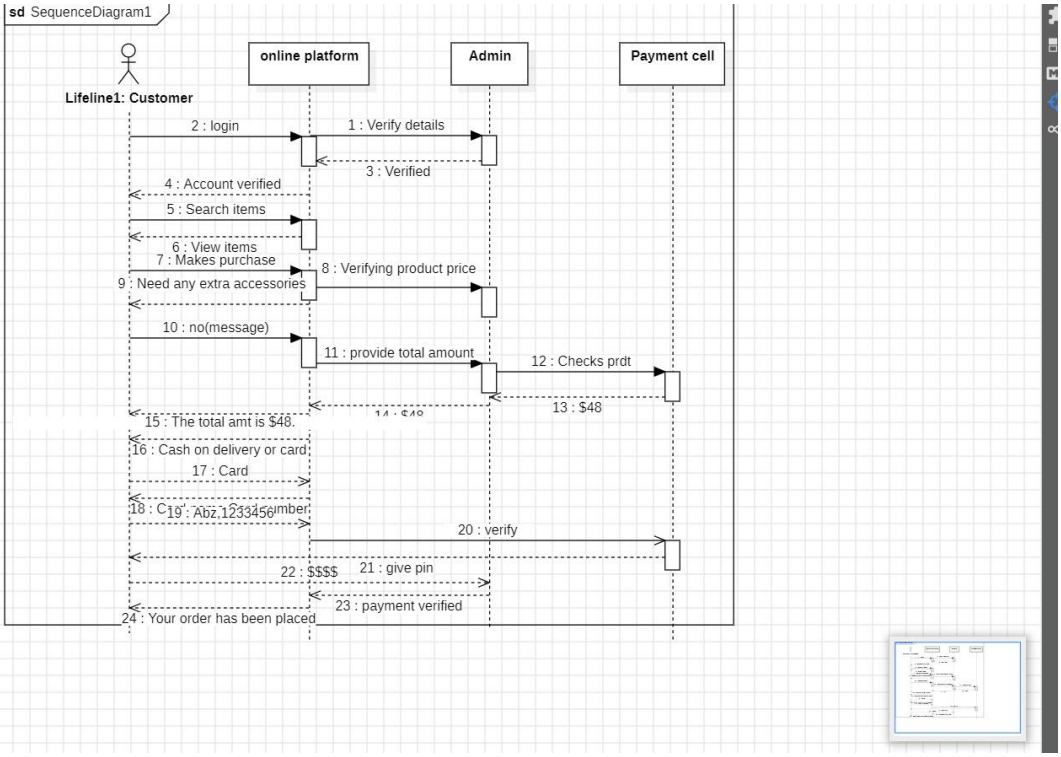
2.a) Use Case Diagram:



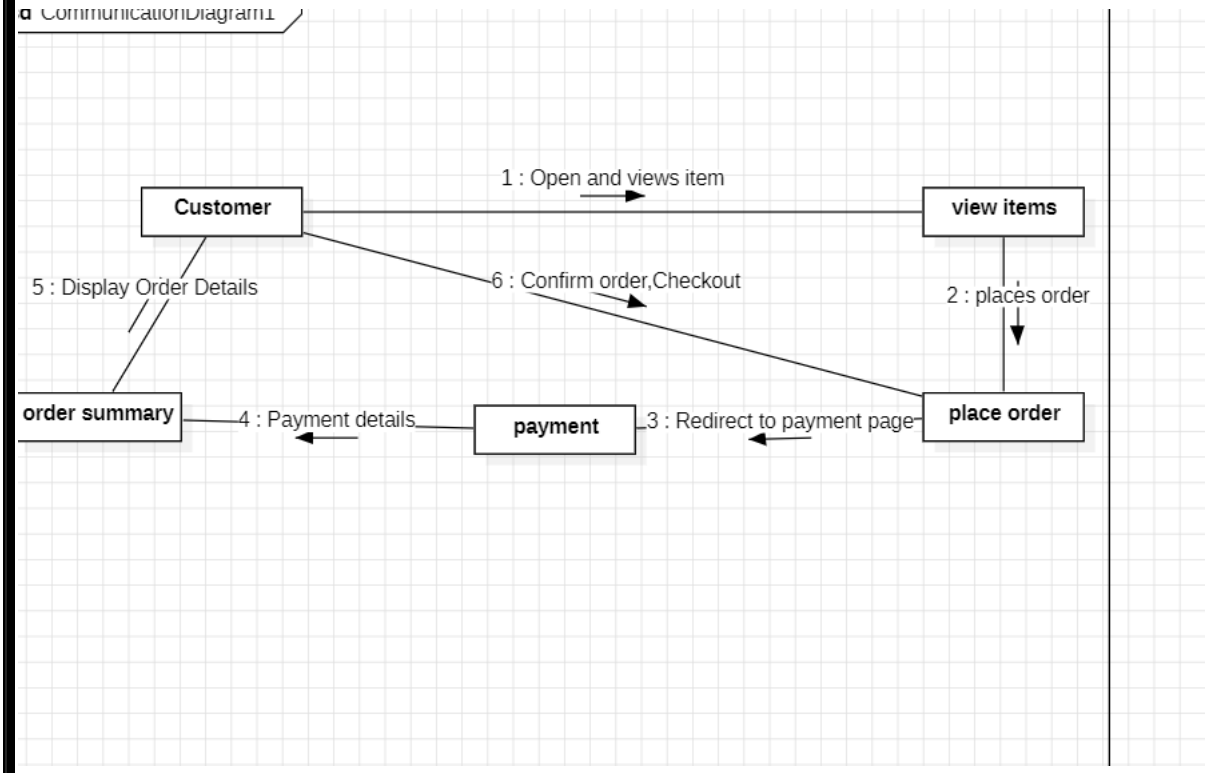
2.b) Class Diagram:



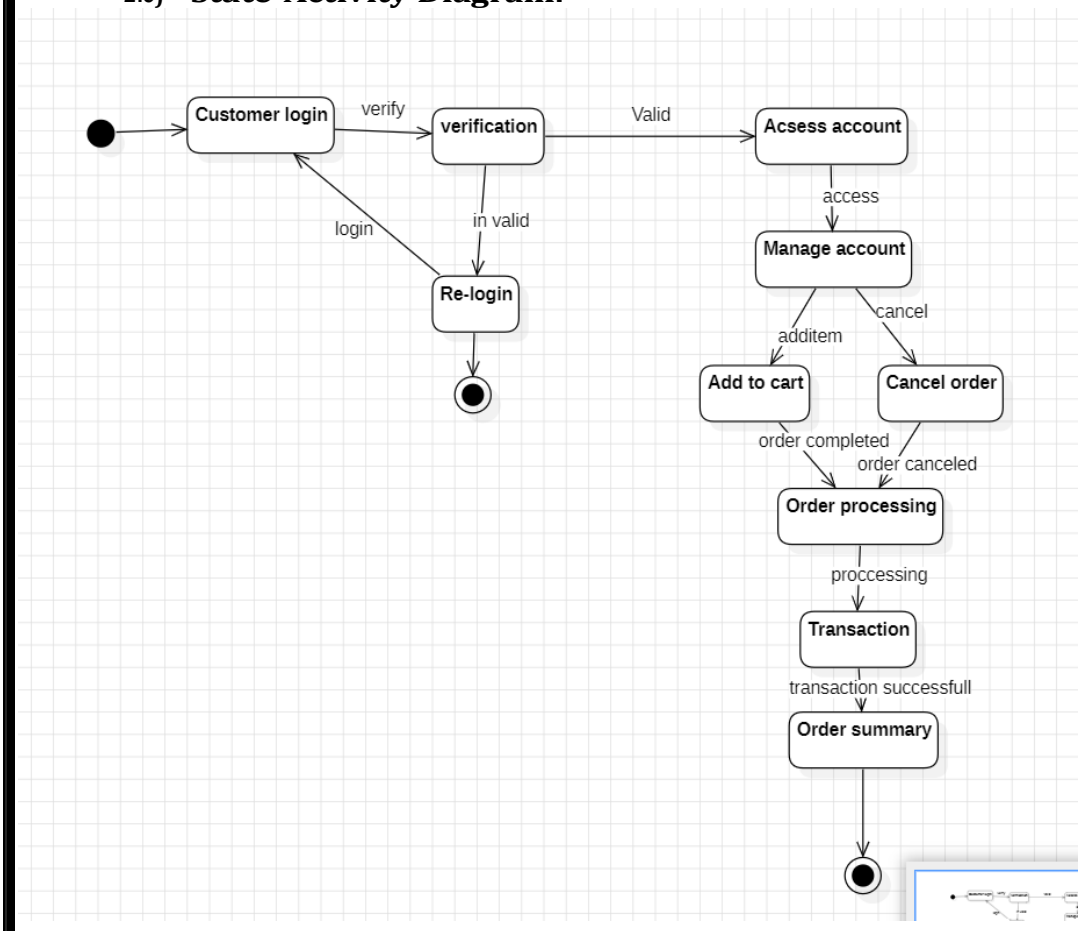
2.c) Sequence Diagram:



2.d) Communication Diagram:



2.e) State-Activity Diagram:



3. Basic Java Programs

3.a) 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.print("Choose operation (+, -, *, /): ");
        char op = sc.next().charAt(0);

        switch (op) {
            case '+': System.out.println("Result: " + (num1 + num2)); break;
            case '-': System.out.println("Result: " + (num1 - num2)); break;
            case '*': System.out.println("Result: " + (num1 * num2)); break;
            case '/':
                if (num2 != 0)
                    System.out.println("Result: " + (num1 / num2));
                else
                    System.out.println("Division by zero not allowed!");
                break;
            default: System.out.println("Invalid operator");
        }
        sc.close();
    }
}
```

Output:

```
C:\Users\Nagarajan\.jdks\openjdk-23.0.1\bin\java.exe
Enter first number: 23
Enter second number: 12
Choose operation (+, -, *, /): +
Result: 35.0
```

3.b) Even, Odd :

Code:

```
import java.util.Scanner;

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

Output:

```
C:\Users\Nagarajan\.jdks\openjdk-23.0.1\bin\java.exe
Enter a number: 13
13 is Odd

Process finished with exit code 0
```

3.c) Factorial:

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, i = 1;

        do {
            fact *= i;
            i++;
        } while (i <= num);

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

Output:

```
C:\Users\Nagarajan\.jdks\openjdk-23.0.1\bin\java.exe
Enter a number: 5
Factorial: 120

Process finished with exit code 0
```

3.d) 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 n = sc.nextInt();

    int a = 0, b = 1, sum;
```

```
        System.out.print("Fibonacci Series: " + a + " " + b);

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

        sc.close();
    }
}
```

Output;

```
C:\Users\Nagarajan\.jdks\openjdk-23.0.1\bin\java.exe
Enter the number of terms: 5
Fibonacci Series: 0 1 1 2 3
Process finished with exit code 0
```

3.e) Number Check:

Code:

```
import java.util.Scanner;

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

        if (num > 0)
            System.out.println("Positive Number");
        else if (num < 0)
            System.out.println("Negative Number");
        else
            System.out.println("Zero");

        sc.close();
    }
}
```

Output:

```
C:\Users\Nagarajan\.jdks\openjdk-23.0.1\bin\java.exe
Enter a number: 12
Positive Number

Process finished with exit code 0
```

3.f) Prime Check:

Code:

```
import java.util.Scanner;

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

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

```
C:\Users\Nagarajan\.jdks\openjdk-23.0.1\bin\java.exe
Enter a number: 12
12 is not a Prime Number

Process finished with exit code 0
```

3.g) Print Number:

Code:

```
import java.util.Scanner;

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

        for (int i = 1; i <= n; i++) {
            System.out.print(i + " ");
        }

        sc.close();
    }
}
```

Output:

```
C:\Users\Nagarajan\.jdks\openjdk-23.0.1\bin\java.exe
Enter a number: 12
1 2 3 4 5 6 7 8 9 10 11 12
Process finished with exit code 0
```

3.h) Reverse 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 rev = 0;

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

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

Output:

```
C:\Users\Nagarajan\.jdks\openjdk-23.0.1\bin\java.exe
Enter a number: 123
Reversed Number: 321

Process finished with exit code 0
```

3.i) SumNatural:

Code:

```
import java.util.Scanner;

public class SumNatural {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int n = sc.nextInt();
        int sum = 0, i = 1;

        while (i <= n) {
            sum += i;
            i++;
        }

        System.out.println("Sum = " + sum);
        sc.close();
    }
}
```

Output:

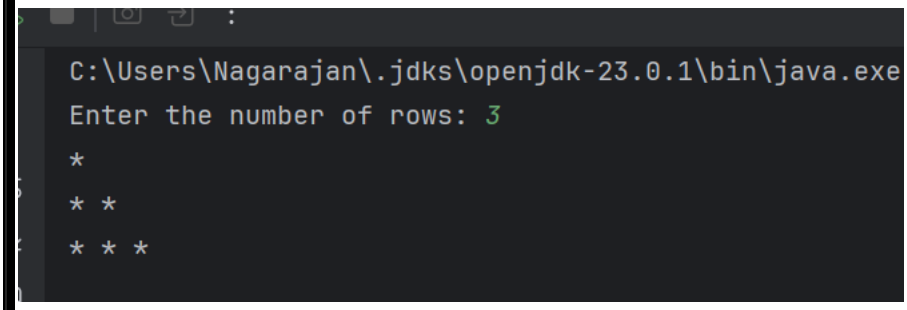
```
C:\Users\Nagarajan\.jdks\openjdk-23.0.1\bin\java.exe
Enter a number: 12
Sum = 78

Process finished with exit code 0
```

3.j) Triangular Pattern:

Code:

```
import java.util.Scanner;

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

        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }

        sc.close();
    }
}
```

Output:



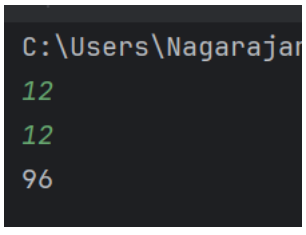
Inheritance

Single inheritance:

Code:

```
import java.util.Scanner;
public class Polyw {

   public static void main(String[] args){
      Scanner s =new Scanner(System.in);
      int a=s.nextInt();
      int b=s.nextInt();
      hlo ob = new bye();
      ob.area(a , b);
   }

}
class hlo{
   int x=2;
   int y=2;
   int z=2;

    void area(int x,int y){
      System.out.println(9*y);
   }
  void area(int x,int y,int z){
       int vol=x*y*z;
      System.out.println(vol);
   }
}
class bye extends hlo{

   void area(int x, int y) {
      System.out.println(8*y);
   }
   void area(int x, int y, int z) {

   }
}
```

screen shot:

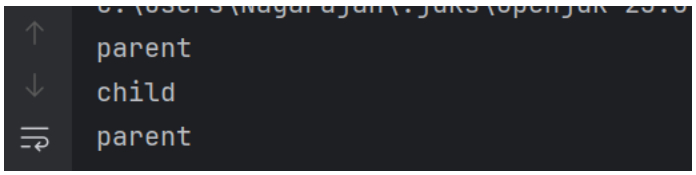


4 b}

Code:

```
public class A {
  static void mani(){
     System.out.println("parent");
   }
   public static void main(String[] args) {
     A a=new A();
     B b=new B();
     A ab=new B();
     a.mani();
     b.mani();
     ab.mani();
   }
}
class B extends A{
  static void mani(){
     System.out.println("child");
   }
}
```

screen shot:



Multiple inhertance

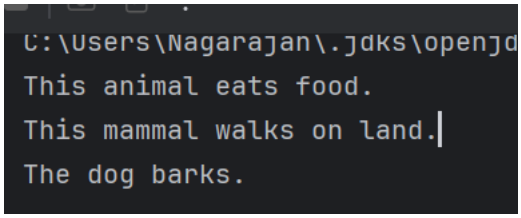
5 a}

Code:

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

// Derived class
class Mammal extends Animal {
   void walk() {
     System.out.println("This mammal walks on land.");
   }
}

// Further derived class
class Dog extends Mammal {
   void bark() {
     System.out.println("The dog barks.");
   }
```

```
}

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

    // Calling methods from all levels of inheritance
    dog.eat();  // From Animal class
    dog.walk();  // From Mammal class
    dog.bark();  // From Dog class
  }
}
```

Screen shot



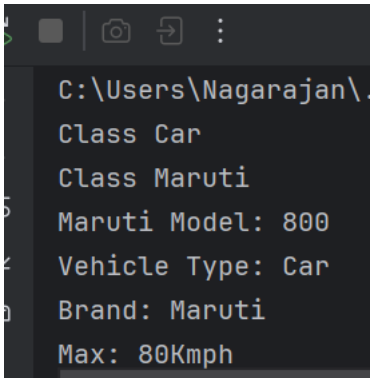
5 b}

Code:

```
class Car{
  public Car()
  {
    System.out.println("Class Car");
  }
  public void vehicleType()
  {
    System.out.println("Vehicle Type: Car");
  }
}
class Maruti extends Car{
  public Maruti()
  {
    System.out.println("Class Maruti");
  }
  public void brand()
  {
    System.out.println("Brand: Maruti");
  }
  public void speed()
  {
    System.out.println("Max: 90Kmph");
  }
}
public class Maruthi800 extends Maruti{

  public Maruthi800()
  {
    System.out.println("Maruti Model: 800");
  }
```

```
  public void speed()
  {
    System.out.println("Max: 80Kmph");
  }
  public static void main(String args[])
  {
    Maruthi800 obj=new Maruthi800();
    obj.vehicleType();
    obj.brand();
    obj.speed();
  }
}
```

screenshot:



Heirarical

6 a}

Code:

```
import java.util.Scanner;
class shape{
  int length,breath,height;
  void length(int length){
  this.length=length;
  }
  void breath(int breath){
    this.breath=breath;
  }
  void height(int height){
    this.height=height;
  }
}
class area extends shape{
  void area(){
    int area =length*breath;
    System.out.println("the area of the given parameter is:"+area);
  }
}
class volume extends shape{
  void volume(){
    int vol=length*breath*height;
```

```
    System.out.println("the volume of the given parameter is:"+vol);
  }
}

public class mains {
  public static void main(String[] args){
    Scanner inp=new Scanner(System.in);
    System.out.println("enter length:");
    int l=inp.nextInt();
    System.out.println("enter breath:");
    int b=inp.nextInt();
    System.out.println("enter height:");
    int h=inp.nextInt();
    area a=new area();
    volume v=new volume();
    a.length(l);
    a.breath(b);
    a.area();
    v.height(h);
    v.breath(b);
    v.length(l);
    v.volume();
  }
}
```

screen shot:

```
C:\Users\Nagarajan\.jdks\openjdk-23.0.1\bin\java.exe --
enter length:
12
enter breath:
12
enter height:
11
the area of the given parameter is:144
the volume of the given parameter is:1584
```

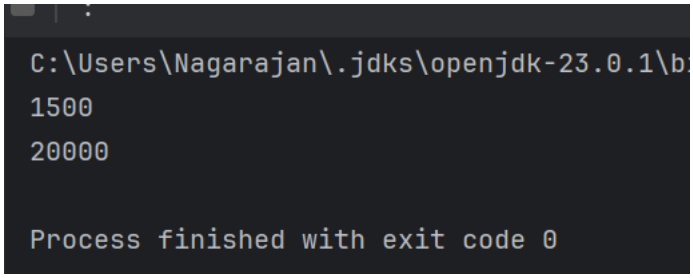
6 b}

Code:

```
public class bank {
  public int getbalance(){
    return 0;
  }
  public static void main(String[] args) {
    bank n=new banka();
    bankb b=new bankb();
    n.getbalance();
    b.getbalance();
  }
}
```

```
class banka extends bank{
  public int getbalance(){
    System.out.println(1500);
    return 1500;
  }
}
class bankb extends bank{
  public int getbalance() {
    System.out.println(20000);
    return 20000;
  }
}
```

screenshot:



Hybrid

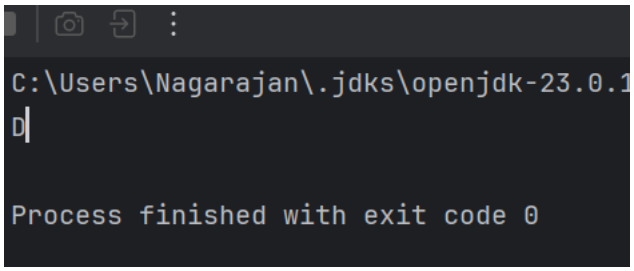
7 a}

Code:

```
class C
{
  public void disp()
  {
    System.out.println("C");
  }
}

class A extends C
{
  public void disp()
  {
    System.out.println("A");
  }
}

class B extends C
{
  public void disp()
  {
    System.out.println("B");
  }

}

class D extends A
```

```
{
   public void disp()
   {
      System.out.println("D");
   }
   public static void main(String args[]){

      D obj = new D();
      obj.disp();
   }
}
```

screen shot:



7 b}

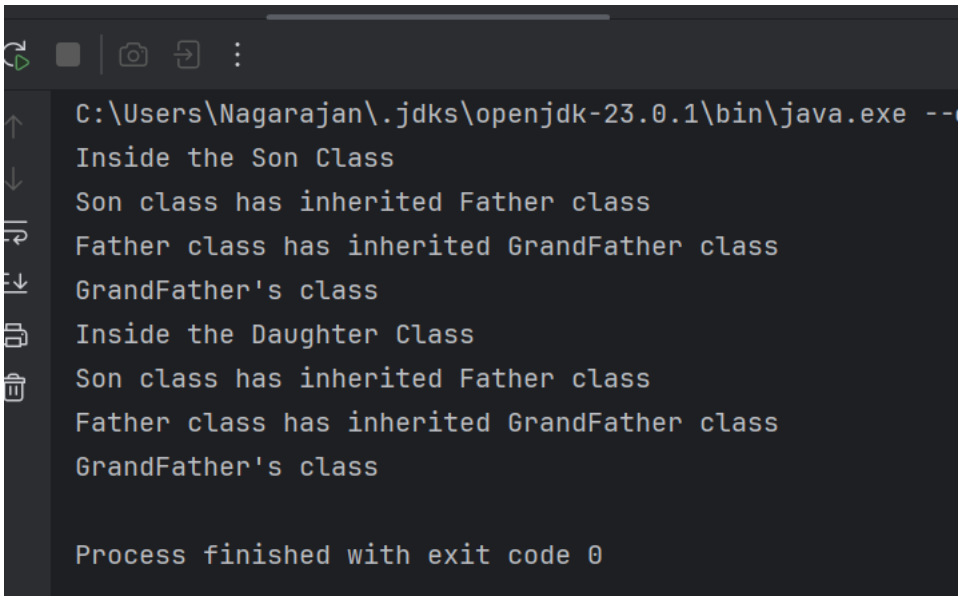
Code:

```
class GrandFather
{
  public void printGrandFather()
  {
     System.out.println("GrandFather's class");
  }
}

class Father extends GrandFather
{
  public void printFather()
  {
     System.out.println("Father class has inherited GrandFather class");
  }
}

class Son extends Father
{
  public Son()
  {
```

```
    System.out.println("Inside the Son Class");
  }

  public void printSon()
  {
    System.out.println("Son class has inherited Father class");
  }
}

class Daughter extends Father
{
  public Daughter()
  {
    System.out.println("Inside the Daughter Class");
  }

  public void printDaughter()
  {
    System.out.println("Son class has inherited Father class");
  }
}

public class HybridInheritance
{
  public static void main(String[]args)
  {
    Son obj = new Son();
    obj.printSon();   // Accessing Son class method
    obj.printFather();   // Accessing Father class method
    obj.printGrandFather();   // Accessing GrandFather class method

    Daughter obj2 = new Daughter();
    obj2.printDaughter();   // Accessing Daughter class method
    obj2.printFather();      // Accessing Father class method
    obj2.printGrandFather();    // Accessing GrandFather class method
```

```
 }
}
```

Screen shot:



Polymorphism

Constructor

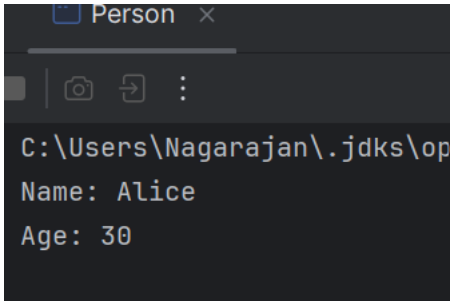
8 a}

Code:

```
// Define a class named 'Person'
class Person {
  // Instance variables
  String name;
  int age;

  // Constructor with parameters
  public Person(String name, int age) {
    this.name = name;
    this.age = age;
  }

  // Method to display the person's details
  public void displayDetails() {
    System.out.println("Name: " + name);
    System.out.println("Age: " + age);
  }

  // Main method to run the program
  public static void main(String[] args) {
    // Create an object of the Person class using the constructor
    Person person1 = new Person("Alice", 30);

    // Display the details of the person
    person1.displayDetails();
```

```
  }
}
```

Screenshot:



Constructor overriding

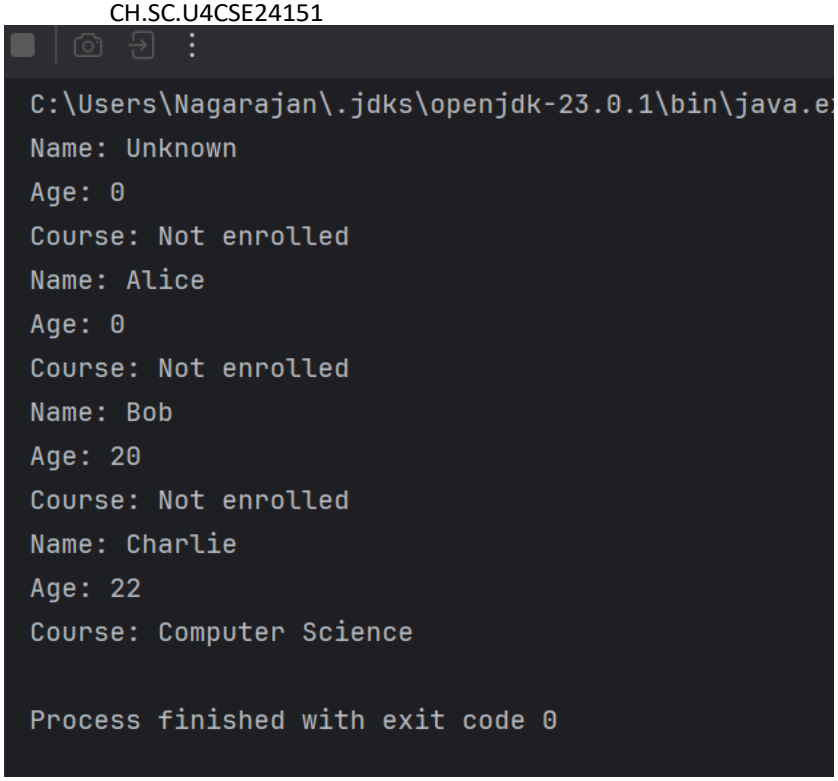
9 a}

Code:

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

  // Default constructor
  public Student() {
    this.name = "Unknown";
    this.age = 0;
    this.course = "Not enrolled";
  }

  // Constructor with one parameter
  public Student(String name) {
    this.name = name;
    this.age = 0;
    this.course = "Not enrolled";
  }

  // Constructor with two parameters
  public Student(String name, int age) {
    this.name = name;
    this.age = age;
    this.course = "Not enrolled";
  }
```

```
  // Constructor with three parameters
  public Student(String name, int age, String course) {
    this.name = name;
    this.age = age;
    this.course = course;
  }

  // Method to display student details
  public void display() {
    System.out.println("Name: " + name);
    System.out.println("Age: " + age);
    System.out.println("Course: " + course);
  }

  public static void main(String[] args) {
    // Creating objects using different constructors
    Student student1 = new Student();
    Student student2 = new Student("Alice");
    Student student3 = new Student("Bob", 20);
    Student student4 = new Student("Charlie", 22, "Computer Science");

    // Displaying student details
    student1.display();
    student2.display();
    student3.display();
    student4.display();
  }
}
```

Screen shot :



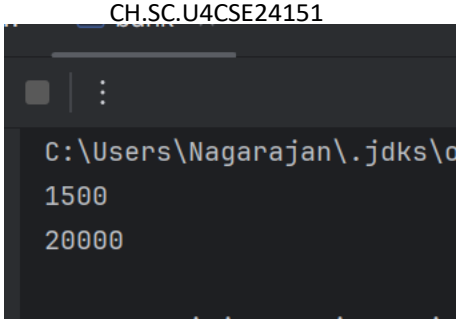
Method overloding

10 a}

Code:

```
public class bank {
   public int getbalance(){
      return 0;
   }
   public static void main(String[] args) {
      bank n=new banka();
      bankb b=new bankb();
      n.getbalance();
      b.getbalance();
   }
}
class banka extends bank{
   public int getbalance(){
      System.out.println(1500);
      return 1500;
   }
}
class bankb extends bank{
   public int getbalance() {
      System.out.println(20000);
      return 20000;
   }
}
```

screen shot:

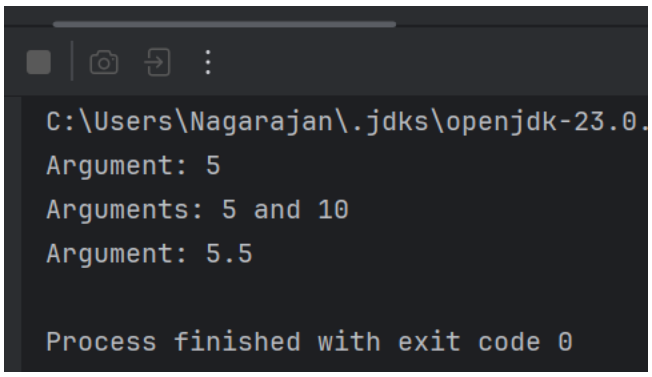


10 b}

Code:

```
public class OverloadExample {

    // Method with one integer parameter
    public void display(int a) {
        System.out.println("Argument: " + a);
    }

    // Method with two integer parameters
    public void display(int a, int b) {
        System.out.println("Arguments: " + a + " and " + b);
    }

    // Method with one double parameter
    public void display(double a) {
        System.out.println("Argument: " + a);
    }

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

        // Calling the method with different parameters
        obj.display(5);
        obj.display(5, 10);
        obj.display(5.5);
    }
}
```

screen shot:



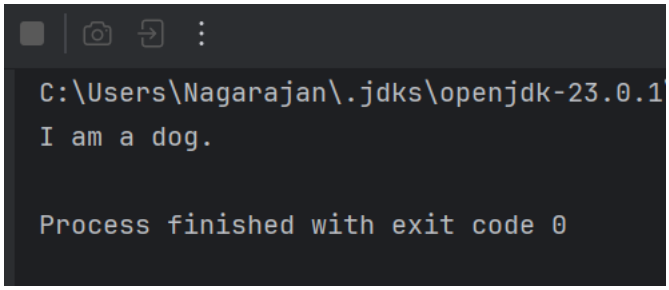
Method over riding

11 a}

Code:

```
class Animal {
  public void displayInfo() {
    System.out.println("I am an animal.");
  }
}

class Dog extends Animal {
  @Override
  public void displayInfo() {
    System.out.println("I am a dog.");
  }
}

class nv {
  public static void main(String[] args) {
    Dog d1 = new Dog();
    d1.displayInfo();
  }
}
```

screen shot:



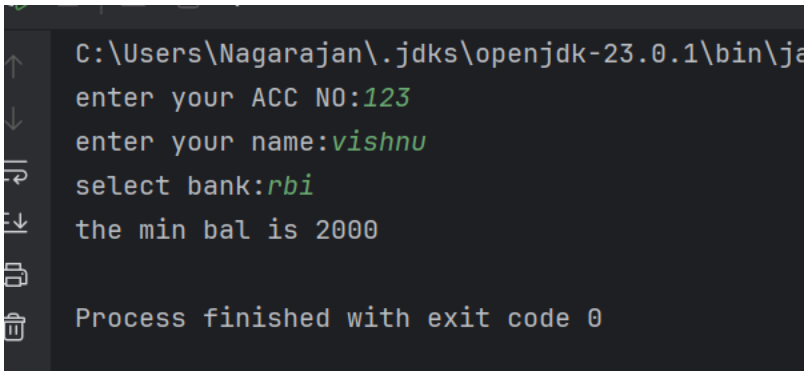
11 b}

Code:

```
import java.util.Scanner;
public class Rbi {
int minbal=2000;
void minbalance(){
System.out.println("the min bal is "+minbal);}
void minwidthdraw(){}
public static void main(String[] args){
Rbi rbi=new Rbi();
SBI sbi=new SBI();
PNB pnb=new PNB();
account c=new account();
c.acc();
Scanner s =new Scanner(System.in);
System.out.print("select bank:");
String sb=s.nextLine();
sb.toLowerCase();
if (sb.equals("rbi")){
rbi.minbalance();
```

```
}
else if (sb.equals("sbi")){
sbi.minbalance();
}
else if (sb.equals("pnb")){
pnb.minbalance();
}
}
}
class SBI extends Rbi{
void minbalance(){
minbal=1500;
System.out.println("the min bal is "+minbal);
}
}
class PNB extends Rbi{
void minbalance(){
minbal=1000;
System.out.println("the min bal is "+minbal);
}
}
class customer extends Rbi{
void cus(){
Scanner in =new Scanner(System.in);
System.out.print("enter your name:");
String name=in.nextLine();
}
}
class account extends customer{
void acc(){
Scanner n=new Scanner(System.in);
System.out.print("enter your ACC NO:");
int accno=n.nextInt();
super.cus();
}
}
```

Screen shot:



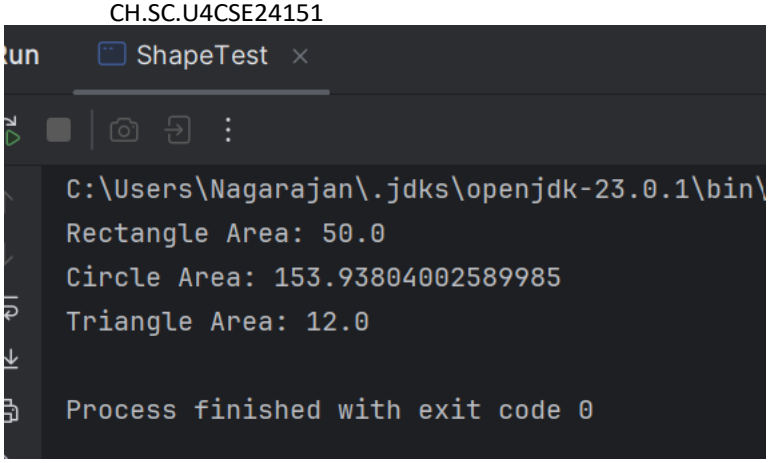
Interface:

12 a}

Code:

```
interface Shape {
   double getArea();
}
class Rectangle implements Shape {
   private double width;
   private double height;

   public Rectangle(double width, double height) {
      this.width = width;
      this.height = height;
   }
   public double getArea() {
      return width * height;
   }
}
class Circle implements Shape {
   private double radius;

   public Circle(double radius) {
      this.radius = radius;
   }
   public double getArea() {
      return Math.PI * radius * radius;
   }
}
class Triangle implements Shape {
   private double base;
   private double height;

   public Triangle(double base, double height) {
      this.base = base;
      this.height = height;
   }
   public double getArea() {
      return 0.5 * base * height;
   }
}
public class ShapeTest {
   public static void main(String[] args) {
      Shape rectangle = new Rectangle(5, 10);
      Shape circle = new Circle(7);
      Shape triangle = new Triangle(4, 6);

      System.out.println("Rectangle Area: " + rectangle.getArea());
      System.out.println("Circle Area: " + circle.getArea());
      System.out.println("Triangle Area: " + triangle.getArea());
   }
}
```

screenshot:

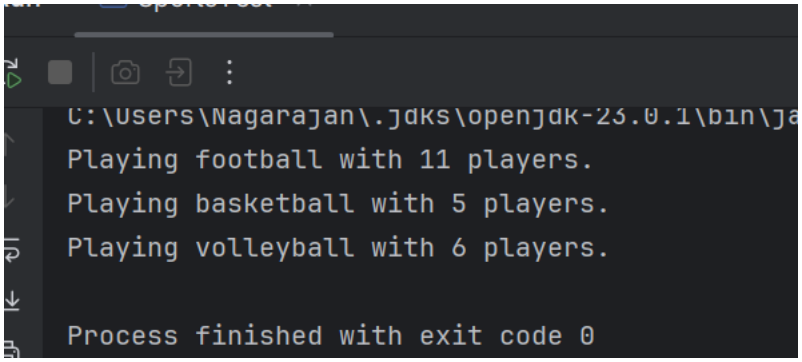


12 b}

Code:

```
interface Playable {
   void play();
}

class Football implements Playable {

   @Override
   public void play() {
      System.out.println("Playing football with 11 players.");
   }
}
class Basketball implements Playable {

   @Override
   public void play() {
      System.out.println("Playing basketball with 5 players.");
   }
}

class Volleyball implements Playable {
   public void play() {
      System.out.println("Playing volleyball with 6 players.");
   }
}
public class SportsTest {
   public static void main(String[] args) {
      Playable football = new Football();
      Playable basketball = new Basketball();
      Playable volleyball = new Volleyball();

      football.play();
      basketball.play();
      volleyball.play();
   }
}
```

screenshot:



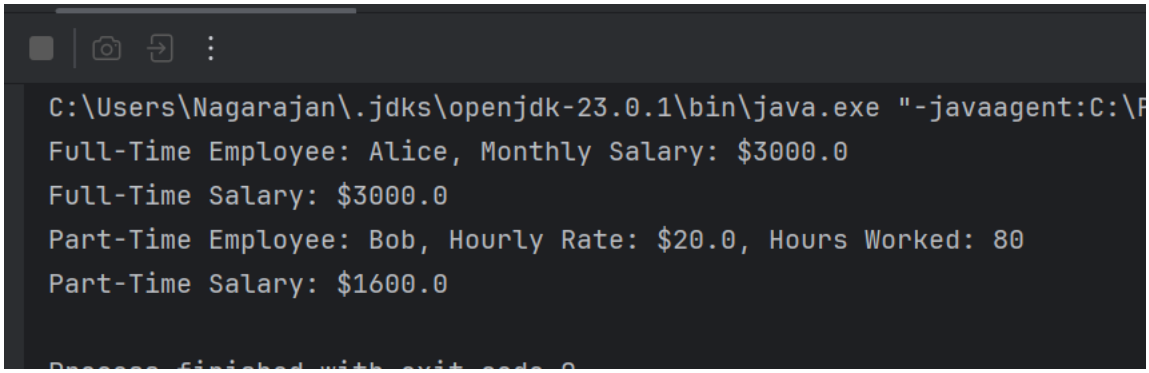
12 c}

Code:

```
// Employee interface
interface Employee {
   double calculateSalary();
   String getDetails();
}

// FullTimeEmployee class
class FullTimeEmployee implements Employee {
   private String name;
   private double monthlySalary;

   public FullTimeEmployee(String name, double monthlySalary) {
      this.name = name;
      this.monthlySalary = monthlySalary;
   }

   @Override
   public double calculateSalary() {
      return monthlySalary;
   }

   @Override
   public String getDetails() {
      return "Full-Time Employee: " + name + ", Monthly Salary: $" + monthlySalary;
   }
}

// PartTimeEmployee class
class PartTimeEmployee implements Employee {
   private String name;
   private double hourlyRate;
   private int hoursWorked;

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

```
  @Override
  public double calculateSalary() {
    return hourlyRate * hoursWorked;
  }

  @Override
  public String getDetails() {
    return "Part-Time Employee: " + name + ", Hourly Rate: $" + hourlyRate + ", Hours Worked: " +
hoursWorked;
  }
}

// Main class to test employee management system
public class EmployeeTest {
  public static void main(String[] args) {
    Employee fullTimeEmp = new FullTimeEmployee("Alice", 3000);
    Employee partTimeEmp = new PartTimeEmployee("Bob", 20, 80);

    System.out.println(fullTimeEmp.getDetails());
    System.out.println("Full-Time Salary: $" + fullTimeEmp.calculateSalary());

    System.out.println(partTimeEmp.getDetails());
    System.out.println("Part-Time Salary: $" + partTimeEmp.calculateSalary());
  }
}
```

Screenshot:



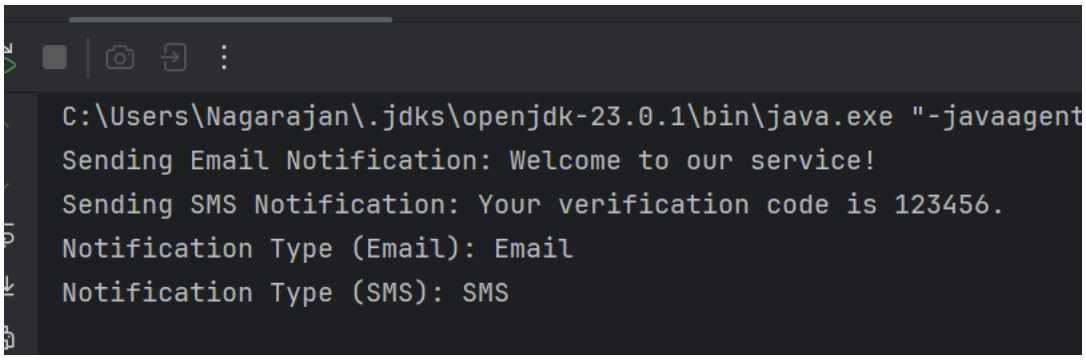
12 d}

Code:

```
// Notifier interface
interface Notifier {
  void sendNotification(String message);
  String getNotificationType();
}

// EmailNotifier class
class EmailNotifier implements Notifier {

  @Override
  public void sendNotification(String message) {
```

```
    System.out.println("Sending Email Notification: " + message);
  }

  @Override
  public String getNotificationType() {
    return "Email";
  }
}

// SMSNotifier class
class SMSNotifier implements Notifier {

  @Override
  public void sendNotification(String message) {
    System.out.println("Sending SMS Notification: " + message);
  }

  @Override
  public String getNotificationType() {
    return "SMS";
  }
}

// Main class to test notification system
public class NotificationTest {

  public static void main(String[] args) {

    Notifier emailNotifier = new EmailNotifier();
    Notifier smsNotifier = new SMSNotifier();

    emailNotifier.sendNotification("Welcome to our service!");
    smsNotifier.sendNotification("Your verification code is 123456.");

    System.out.println("Notification Type (Email): " + emailNotifier.getNotificationType());
    System.out.println("Notification Type (SMS): " + smsNotifier.getNotificationType());
  }
}
```

Screen shot:



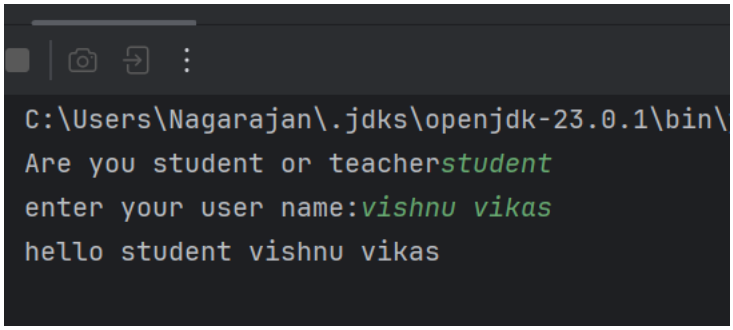
Abstract class

13 a}

Code:

```
import java.util.Scanner;
public class sss{
   public static void main(String[] args){
      Scanner inp=new Scanner(System.in);
      System.out.print("Are you student or teacher");
      String nam=inp.nextLine();
      nam=nam.toLowerCase();
      System.out.print("enter your user name:");
      String nam1=inp.nextLine();
      if (nam.equals("student")){
         en obj=new student(nam1);
         obj.display();
      } else if (nam.equals("teacher")) {
         en obj=new teacher(nam1);
         obj.display();
      }
   }
}
abstract class en{
   abstract void display();
}
class student extends en{
   String a;
   student(String a){
      this.a=a;
   }
   void display(){
      System.out.println("hello student "+a);
   }
}
class teacher extends en{
   String a;
   teacher(String a){
      this.a=a;
   }
   void display(){
      System.out.println("hello teacher "+a);
   }
}
```

screen shot:



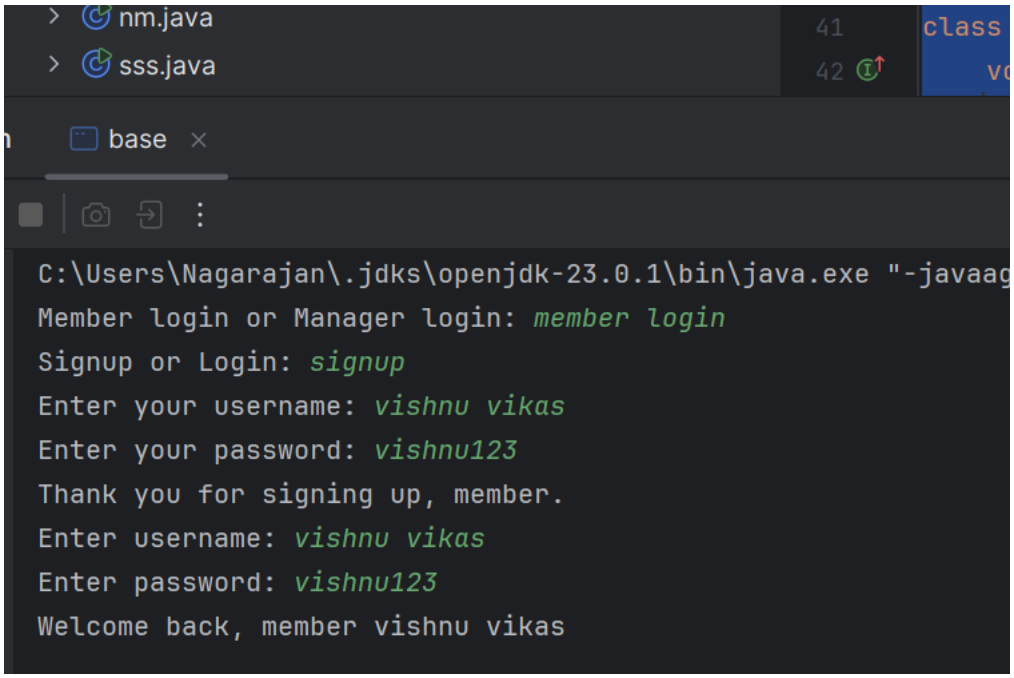
13 b}

Code:

```
import java.util.Scanner;

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

      System.out.print("Member login or Manager login: ");
      String ask = inp.nextLine().toLowerCase();

      System.out.print("Signup or Login: ");
      String akss = inp.nextLine().toLowerCase();

      if (ask.equals("member login")) {
         Member member = new Member();
         if (akss.equals("signup")) {
            member.signup();
         }
         member.login();
      } else if (ask.equals("manager login")) {
         Manager manager = new Manager();
         if (akss.equals("signup")) {
            manager.signup();
         }
         manager.login();
      } else {
         System.out.println("Invalid option.");
      }
      inp.close();
   }
}

abstract class Hello {
   protected String usr;
   protected String psw;
   protected Scanner inp = new Scanner(System.in);

   abstract void signup();
   abstract void login();
}
```

```
class Member extends Hello {
  void signup() {
    System.out.print("Enter your username: ");
    usr = inp.nextLine();
    System.out.print("Enter your password: ");
    psw = inp.nextLine();
    System.out.println("Thank you for signing up, member.");
  }
  void login() {
    System.out.print("Enter username: ");
    String user = inp.nextLine();
    System.out.print("Enter password: ");
    String pswd = inp.nextLine();

    if (user.equals(usr) && pswd.equals(psw)) {
      System.out.println("Welcome back, member " + user);
    } else {
      System.out.println("Incorrect username or password.");
    }
  }
}

class Manager extends Hello {
  void signup() {
    System.out.print("Enter your username: ");
    usr = inp.nextLine();
    System.out.print("Enter your password: ");
    psw = inp.nextLine();
    System.out.println("Thank you for signing up, manager.");
  }
  void login() {
    System.out.print("Enter username: ");
    String user = inp.nextLine();
    System.out.print("Enter password: ");
    String pswd = inp.nextLine();

    if (user.equals(usr) && pswd.equals(psw)) {
      System.out.println("Welcome back, manager " + user);
    } else {
      System.out.println("Incorrect username or password.");
    }
  }
}
```

screen shot:



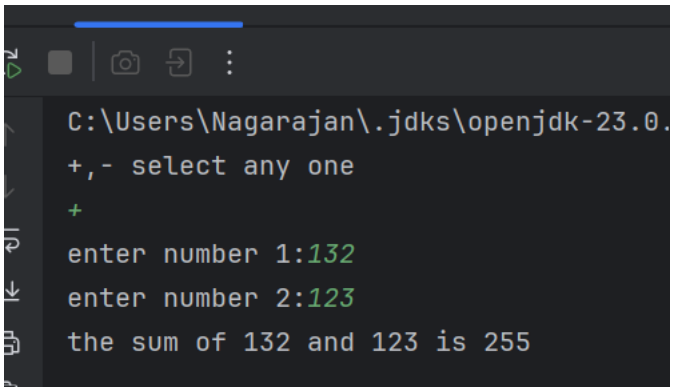
13 c}

Code:

```
import java.util.Scanner;
public class bse {
   public static void main(String[] args) {
      Scanner input=new Scanner(System.in);
      System.out.println("+,- select any one");
      String gett=input.nextLine();
      if(gett.equals("+")){
         disp v=new add();
         v.calc();
      } else if (gett.equals("-")) {
         disp n=new sub();
         n.calc();
      }
   }
}
abstract class disp{
   protected int a,b;
   protected Scanner input=new Scanner(System.in);
   abstract void calc();
}
class add extends disp{
   void calc(){
      System.out.print("enter number 1:");
      a=input.nextInt();
      System.out.print("enter number 2:");
      b=input.nextInt();
      System.out.println("the sum of "+a+" and "+b+" is "+(a+b));
```

```
  }
}
class sub extends disp{
  void calc(){
    System.out.print("enter number 1:");
    a=input.nextInt();
    System.out.print("enter number 2:");
    b=input.nextInt();
    System.out.println("the difference is "+(a-b));
  }
}
```

screenshot:

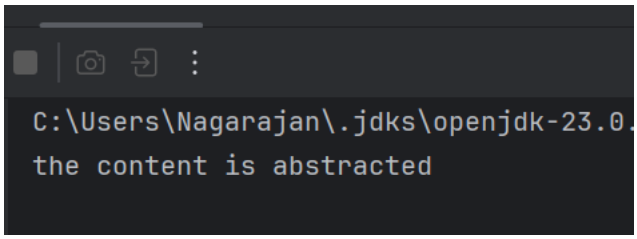


13 d}

Code:

```
public class nm {
  public static void main(String[] args) {
    bb nnn=new bb();
    nnn.bk();
  }
}
abstract class neww{
  abstract void bk();
}
class bb extends neww {
  void bk(){
    System.out.println("the content is abstracted");
  }
}
```

screen shot:



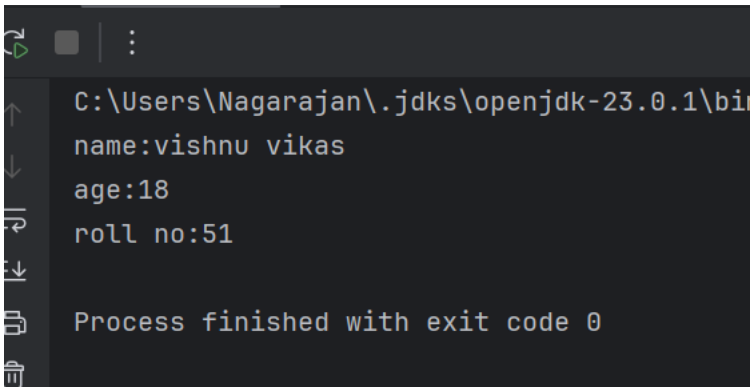
Encapsulation

14 a}

Code:

```
public class encap{
public static void main(String[] args){
cdd obj = new cdd();
obj.setroll(51);
obj.setname("vishnu vikas");
obj.setage(18);
System.out.println("name:"+obj.getname());
System.out.println("age:"+obj.getage());
System.out.println("roll no:"+obj.getroll());


}
}
class cdd{
private int age;
private String name;
private int roll;
public void setroll(int ns){
roll = ns;
}
public void setname(String ns){
name = ns;
}
public void setage(int ns){
age = ns;
}
public int getroll(){
return roll;
}
public int getage(){
return age;
}
public String getname(){
return name;
}
}
```

screen shot:

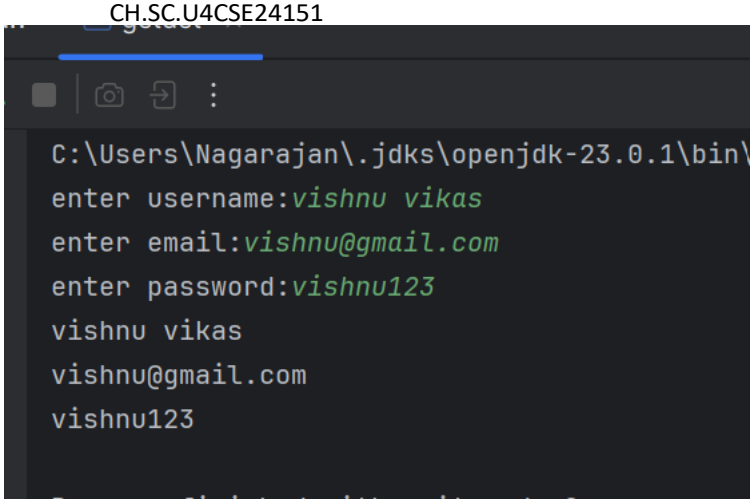


14 b}

Code:

```
import java.util.Scanner;
public class getdet{
public static void main(String[] args){
Scanner inp = new Scanner(System.in);
System.out.print("enter username:");
String usr=inp.nextLine();
System.out.print("enter email:");
String mail=inp.nextLine();
System.out.print("enter password:");
String pss=inp.nextLine();
det obj = new det();
obj.setuser(usr);
obj.setemail(mail);
obj.setpass(pss);
System.out.println(obj.getuser());
System.out.println(obj.getemail());
System.out.println(obj.getpass());
}
}
class det{
private String user;
private String email;
private String pass;

public void setuser(String ns){
user=ns;
}
public void setemail(String ns){
email=ns;
}
public void setpass(String ns){
pass=ns;
}
public String getuser(){
return user;
}
public String getemail(){
return email;
}
public String getpass(){
return pass;
}
}
```

screen shot:

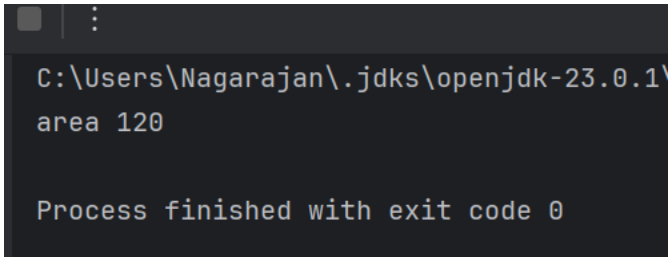


14 c}

Code:

```
public class area{
public static void main(String[] args){
arw ob = new arw();
ob.setlen(12);
ob.setbre(10);
System.out.println("area "+(ob.getlen()*ob.getbre()));
}
}
class arw{
private int len;
private int bre;
public void setlen(int ns){
len =ns;
}
public void setbre(int ns){
bre =  ns;
}
public int getlen(){
return len;
}
public int getbre(){
return bre;
}
}
```

screen shot:



14 d}

Code:

```
class Account {
   //private data members
   private long acc_no;
   private String name,email;
   private float amount;
   //public getter and setter methods
   public long getAcc_no() {
      return acc_no;
   }
   public void setAcc_no(long acc_no) {
      this.acc_no = acc_no;
   }
   public String getName() {
      return name;
   }
   public void setName(String name) {
      this.name = name;
   }
   public String getEmail() {
      return email;
   }
   public void setEmail(String email) {
      this.email = email;
   }
   public float getAmount() {
      return amount;
   }
   public void setAmount(float amount) {
      this.amount = amount;
   }
}
//A Java class to test the encapsulated class Account.
public class mnnn {
   public static void main(String[] args) {
      //creating instance of Account class
      Account acc=new Account();
      //setting values through setter methods
      acc.setAcc_no(7560504000L);
      acc.setName("vishnu");
      acc.setEmail("viahu@gmail.com");
      acc.setAmount(500000f);
      //getting values through getter methods
      System.out.println(acc.getAcc_no());
      System.out.println(acc.getName());
      System.out.println(acc.getEmail());
      System.out.println(acc.getAmount());

   }
```

screen shot:

```
7560504000
vishnu
vishnu@gmail.com
500000.0
```

15 a}

Code:

```
package one;
public class cc{
public void mn(String a){
System.out.println(a);
}
public static void main(String[] args){
cc a = new cc();
a.mn("hello");
}
}

import one;
public class demo{
public static void main(String[] args){
cc n= new cc();
n.mn("vishun vikas");
}
}
```

Screen shot:

```
C:\Users\Nagarajan\OneDrive\Documents\java>javac -d . cc.java

C:\Users\Nagarajan\OneDrive\Documents\java>javac demo.java

C:\Users\Nagarajan\OneDrive\Documents\java>java demo
vishun vikas
```

15 b}

Code:

```
  package bb;
```

```
public class Demmo {
 public void sum(int num1, int num2) {
  int result;
  result = num1 + num2;
  System.out.println("the sum of two numbers is:" + result);
 }
}
```

```
import bb.Demmo;
class Tester extends Demmo {
 public static void main(String args[]) {
  Tester obj = new Tester();
  obj.sum(10, 20);
 }
}
```

Screen shot:

```
C:\Users\Nagarajan\OneDrive\Documents\java>javac -d . Demmo.java

C:\Users\Nagarajan\OneDrive\Documents\java>javac Tester.java

C:\Users\Nagarajan\OneDrive\Documents\java>java Tester
the sum of two numbers is:30
```

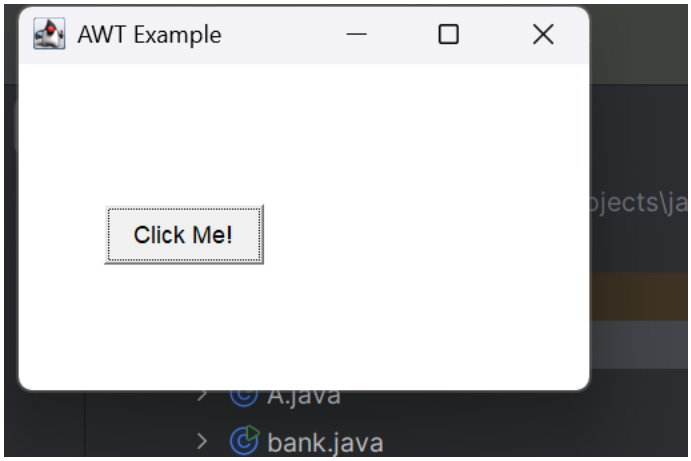
15 c}

Code:

```
import java.awt.*;
import java.awt.event.*;
 public class SimpleAWTApp {
   SimpleAWTApp() {
      Frame frame = new Frame("AWT Example");
      Button button = new Button("Click Me!");

      button.setBounds(50, 100, 80, 30);
      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 SimpleAWTApp();
  }
}
```

screen shot:

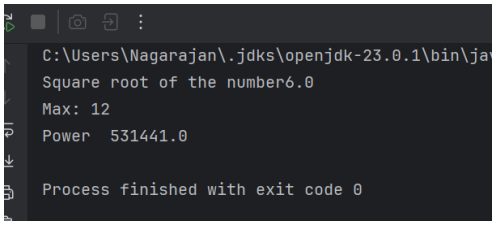


15 d}

Code:

```
import java.lang.*;
public class math {
  public static void main(String[] args) {
    int a=3,b=12;
    System.out.println("Square root of the number" + Math.sqrt(a*b));
    System.out.println("Max: " + Math.max(a, b));
    System.out.println("Power  " + Math.pow(a, b));
  }

}
```

screen shot:



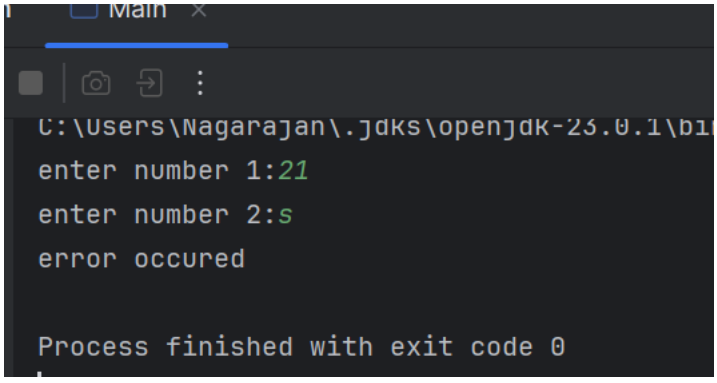
Exceptional handiling :

16 a}

Code:

```
import java.util.*;
public class Main{
  public static void main(String[] args) {


  arthematic n= new arthematic();
  n.add();
  }
}
class arthematic{
  int a,b;
  void add(){
    try{
      Scanner inp= new Scanner(System.in);
      System.out.print("enter number 1:");
      int a=inp.nextInt();
      System.out.print("enter number 2:");
      int b= inp.nextInt();
      System.out.println(a/b);
    }
    catch (InputMismatchException e){
      System.out.println("error occured");
    }
  }
}
```

screen shot:



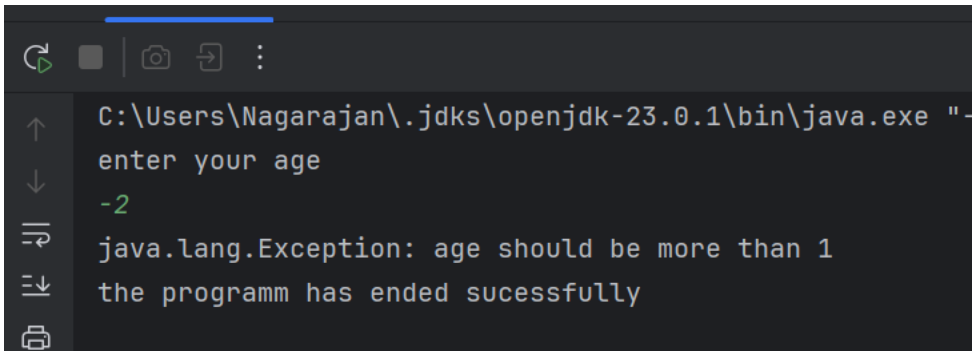
16 b}

Code:

```
import java.util.*;
public class asd {
  public static void main(String[] args) {
    detal n= new detal();
    n.mm();
  }
}
class detal{
```

```
  void mm(){
    try{
    Scanner inp= new Scanner(System.in);
    System.out.println("enter your age");
    int age=inp.nextInt();
    if (age<1){
      throw new Exception("age should be more than 1");
    }
    }
    catch (Exception e){
      System.out.println(e);
    }
    finally {
      System.out.println("the programm has ended sucessfully");
    }
  }
}
```

screen shot:



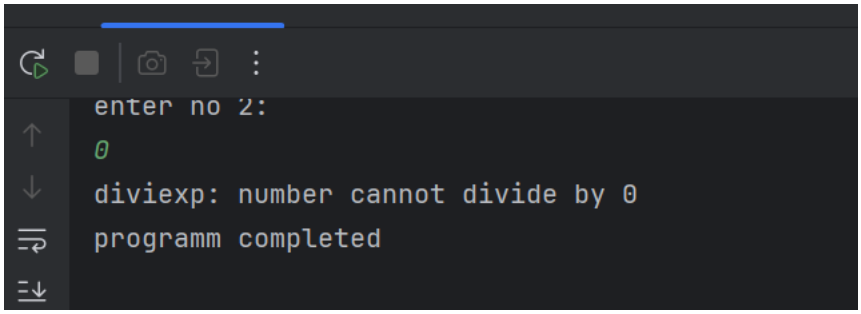
16 c}

Code:

```
import java.util.*;
public class qww {
  public static void main(String[] args) {
    division n= new division();
    n.add();
  }
}
class diviexp extends Exception{
  diviexp(String s){
    super(s);
  }
}
class division{
  int a,b;
  void add(){
    try {
      Scanner in = new Scanner(System.in);
      System.out.println("enter no 1:");
      a = in.nextInt();
      System.out.println("enter no 2:");
      b = in.nextInt();
```

```
        if(b==0){
          throw new diviexp("number cannot divide by 0");
        }
      }
      catch (diviexp e){
        System.out.println(e);
      }
      finally {
        System.out.println("programm completed ");
      }
    }
}
```

screen shot:



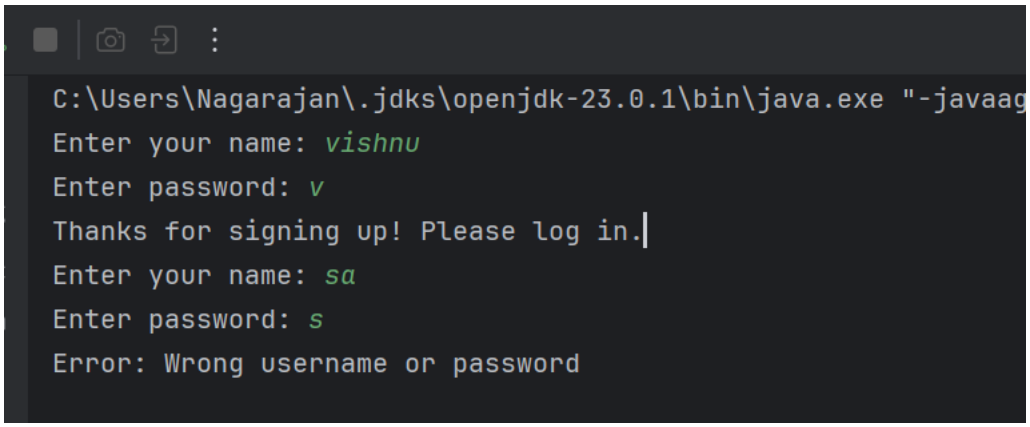
16 d}

Code:

```
import java.util.*;

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

    System.out.print("Enter your name: ");
    String nam = n.nextLine();

    System.out.print("Enter password: ");
    String pass = n.nextLine();
    Login p = new Login();
    p.signup(nam, pass);
  }
}
class SyntaxException extends Exception {
  SyntaxException(String message) {
    super(message);
  }
}

class Login {
  void signup(String nam, String pass) {
    System.out.println("Thanks for signing up! Please log in.");

    try {
      Scanner nn = new Scanner(System.in);
```

```
        System.out.print("Enter your name: ");
        String nm = nn.nextLine();

        System.out.print("Enter password: ");
        String pss = nn.nextLine();

        // Corrected String comparison
        if (!nm.equals(nam) || !pss.equals(pass)) {
          throw new SyntaxException("Wrong username or password");
        }
        System.out.println("Login successful!");
      } catch (SyntaxException e) {
        System.out.println("Error: " + e.getMessage());
      }
    }
}
```

screen shot:



File handiling

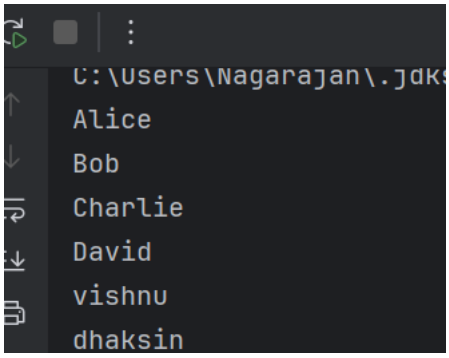
17 a}

Code:

```
import java.io.FileReader;
import java.io.BufferedReader;
public class Main{
  public static void main(String[] args) {
    try{
    FileReader fr = new FileReader("output.txt");
    BufferedReader br=new BufferedReader(fr);
      String v= br.readLine();
      while (v!=null){
        System.out.println(v);
        v= br.readLine();
      }


      System.out.println("succes");
  }
```

```
  catch (Exception e){
    System.out.println("error occured");
  }
  }
}
```

screen shot:



17 B}

Code:

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

import java.util.ArrayList;


public class sup {
  public static void main(String[] args) {
    ArrayList<String> arr =new ArrayList<>();

    try {
      FileReader fr = new FileReader("output.txt");
      BufferedReader br = new BufferedReader(fr);
      String v= br.readLine();

      int a=0;
      while (v!=null){
        System.out.println(v);
        arr.add(v);
        v= br.readLine();
        a+=1;
        System.out.println(a);
      }
    }
    catch (Exception e){
      System.out.println("error occured");
    }
    System.out.println(arr);

  }
}
```

screenshot:

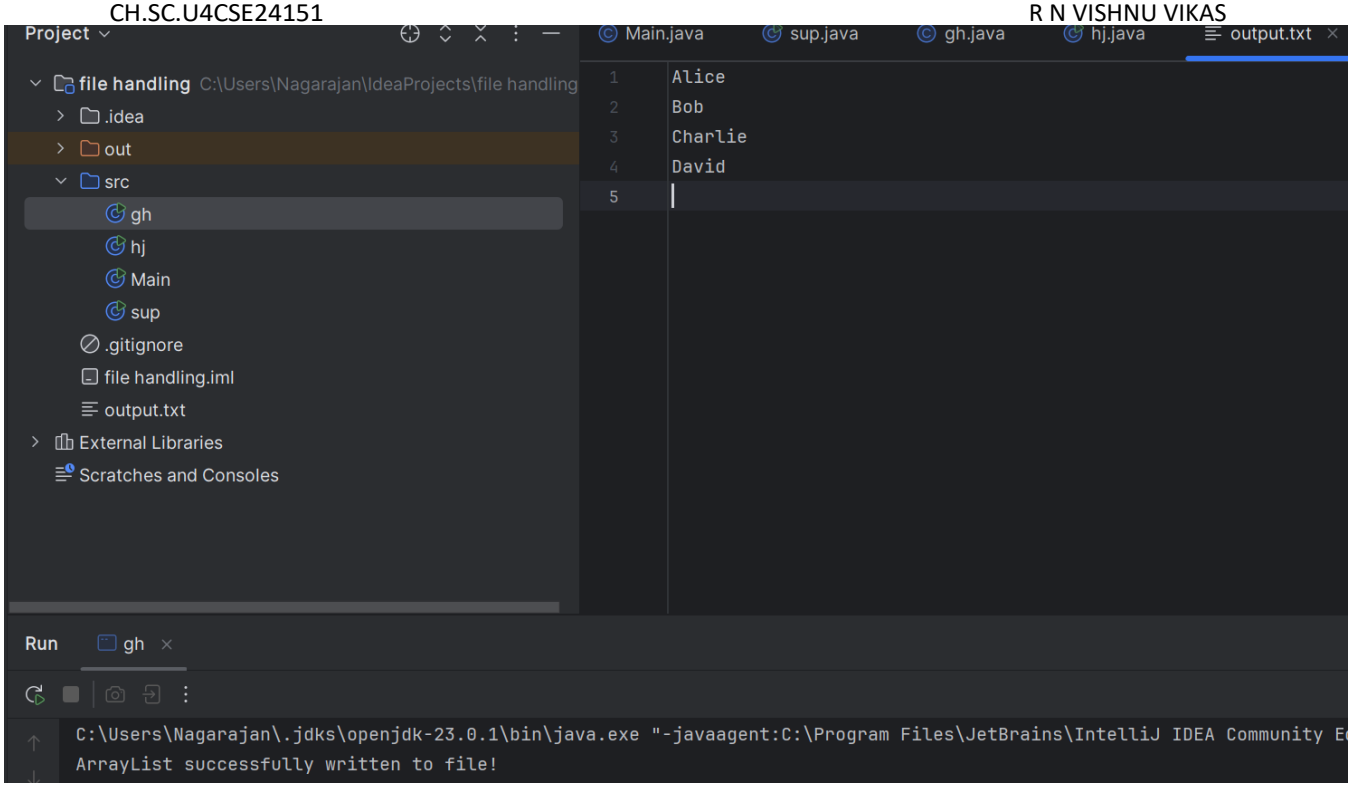
```
djns
9
dfskj
10
sdf
11
fds
12
dfs
13
[Alice, Bob, Charlie, David, vishnu, dhaksin, vishnu, jn, djns, dfskj, sdf, fds, dfs]

Process finished with exit code 0
```

17 c}

Code:

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;

public class gh {
  public static void main(String[] args) {
    // Create an ArrayList of strings
    ArrayList<String> names = new ArrayList<>();
    names.add("Alice");
    names.add("Bob");
    names.add("Charlie");
    names.add("David");
    String filePath = "output.txt";
    try (BufferedWriter writer = new BufferedWriter(new FileWriter(filePath))) {
      for (String name : names) {
        writer.write(name);
        writer.newLine();
      }
      System.out.println("ArrayList successfully written to file!");
    } catch (IOException e) {
      System.out.println("An error occurred: " + e.getMessage());
    }
  }
}
```

screen shot:



17 d}

Code:

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.util.Arrays;
import java.util.Scanner;
public class hj {
   public static void main(String[] args) {
     Scanner inp = new Scanner(System.in);
      System.out.println("how many name you want to enter:");
      int aa=inp.nextInt();
      inp.nextLine();
      String[] arr = new String[aa];
     for (int i=0;i<aa;i++){
        System.out.println("enter name "+(i+1)+":");
        String nam=inp.nextLine();
        arr[i]=nam;
     }
     try {
        FileWriter fw = new FileWriter("output.txt",true);
        BufferedWriter bw = new BufferedWriter(fw);
         for (int j=0;j<aa;j++){
            bw.newLine();
            bw.write(arr[j]);
         }
         bw.close();
        System.out.println("ArrayList successfully written to file!");
     }
    catch (Exception e){
       System.out.println("error");
```

```
    }

      System.out.println(Arrays.asList(arr));
  }
}
```

screen shot: