



SCHOOL OF
COMPUTING

LAB RECORD

23CSE111- Object Oriented Programming

Submitted by

CH.SC.U4CSE24124 -Maddu Raahithya Yadav

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND
ENGINEERING

AMRITA VISHWA VIDYAPEETHAM
AMRITA SCHOOL OF COMPUTING

CHENNAI

March - 2025



SCHOOL OF
COMPUTING

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.U4CSE24124 – Maddu Raahithya Yadav** 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

S.NO	TITLE	PAGE.NO
	UML DIAGRAM	
1.	BANK MANAGEMENT SYSTEM	
	1.a) Use Case Diagram	6
	1.b) Class Diagram	7
	1.c) Sequence Diagram	7
	1.d) Object Diagram	8
	1.e) State Diagram	8
2.	COLLEGE MANAGEMENT SYSTEM	
	2.a) Use Case Diagram	9
	2.b) Class Diagram	10
	2.c) Sequence Diagram	11
	2.d) Object Diagram	12
	2.e) State Diagram	12
3.	BASIC JAVA PROGRAMS	
	3.a) To display details of a student by input	13
	3.b) Printing Pattern by taking input from user.	14
	3.c) To check if the given letter is vowel or not	15
	3.d) Sum of digits of a number	16
	3.e) Reversing a number	17
	3.f) Fibonacci series	18
	3.g) Deposit and Withdrawal	19
	3.h) Grade of student by input	20
	3.i) Area of circle	21
	3.j) Simple Calculator	22
	INHERITANCE	
4.	SINGLE INHERITANCE PROGRAMS	
	4.a) Appliances and washing Machine	23
	4.b) Employee, manager details	24

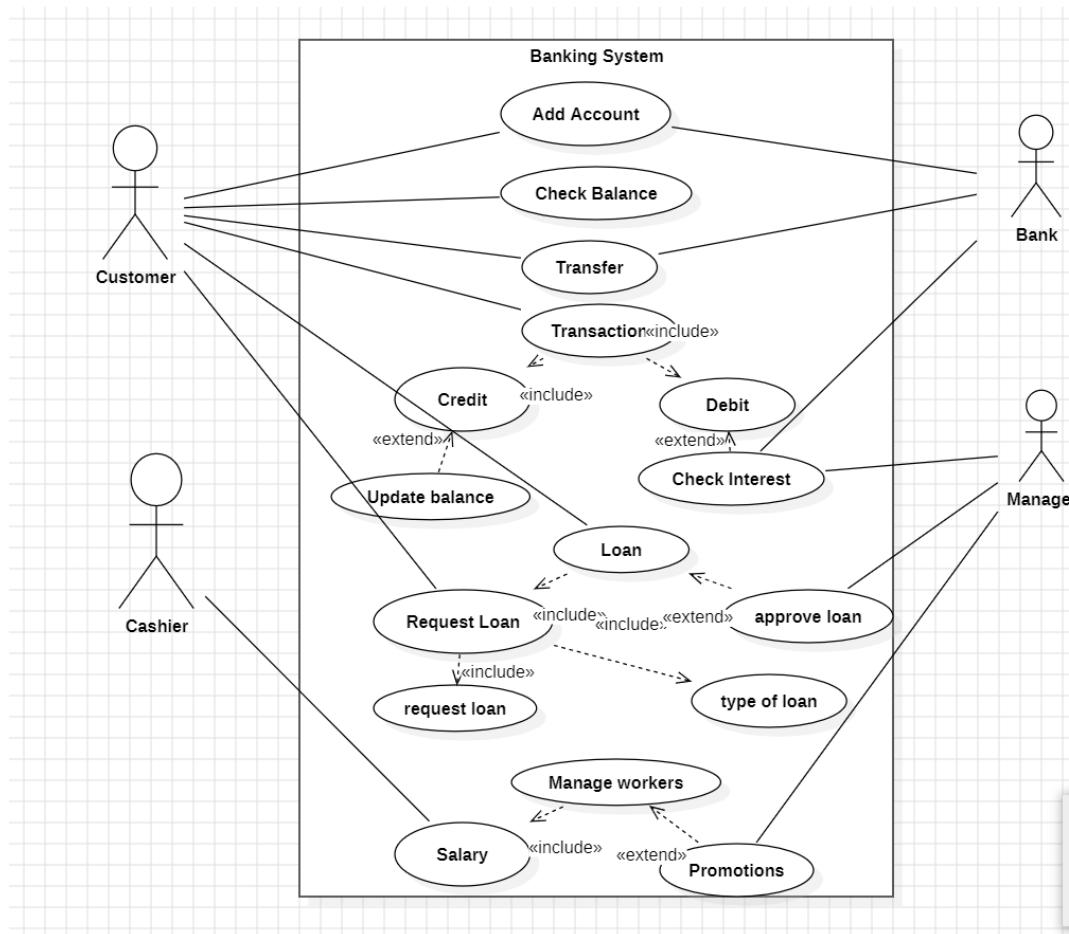
5.	MULTILEVEL INHERITANCE PROGRAMS	
	5.a) Fibonacci series	25-26
	5.b) Dice probability	27-28
6.	HIERARCHICAL INHERITANCE PROGRAMS	
	6.a) Planet Details	29-30
	6.b) Time Converter	31
7.	HYBRID INHERITANCE PROGRAMS	
	7.a) Country,city,district	32-33
	7.b) Shopping cart	34-35
	POLYMORPHISM	
8.	CONSTRUCTOR PROGRAMS	
	8.a) company stock market	36-37
9.	CONSTRUCTOR OVERLOADING PROGRAMS	
	9.a) Player info	38-39
10.	METHOD OVERLOADING PROGRAMS	
	10.a) Power calculator	40
	10.b) Age calculator	41
11.	METHOD OVERRIDING PROGRAMS	
	11.a) Recursive factorial	42
	11.b) Security Mnagement	43
	ABSTRACTION	
12.	INTERFACE PROGRAMS	
	12.a) ATM Machine Interface	44
	12.b) Shapes, polygons	45-46
	12.c) Pyment Interface	47
	12.d) Math Operations	48-49
13.	ABSTRACT CLASS PROGRAMS	
	13.a) Tourism Booking	50-51
	13.b) Home Automation	52-53
	13.c) Package delivary	54-55
	13.d) Chemical Elements	56-57
	ENCAPSULATION	
14.	ENCAPSULATION PROGRAMS	
	14.a) Library books	58-59
	14.b) Movie tickets booking	60-61
	14.c) Online Courses	62-63
	14.d) Devices	64-65
15.	PACKAGES PROGRAMS	
	15.a)User Defined Packages-Hospital Management	66-67
	15.b)User Defined Packages-Library Management	68-69
	15.c)Built - in Package(3 Packages)-Reverse Names	70
	15.d)Built - in Package(3 Packages) – Items List	71

16.	EXCEPTION HANDLING PROGRAMS	
	16.a) User input	72
	16.b) File not found	73
	16.c) Login system	74
	16.d) Divide by Zero	75
17.	FILE HANDLING PROGRAMS	
	17.a) File informatation	76
	17.b) Write to a File	77
	17.c) Read to a File	78
	17.d) Append to a file	79

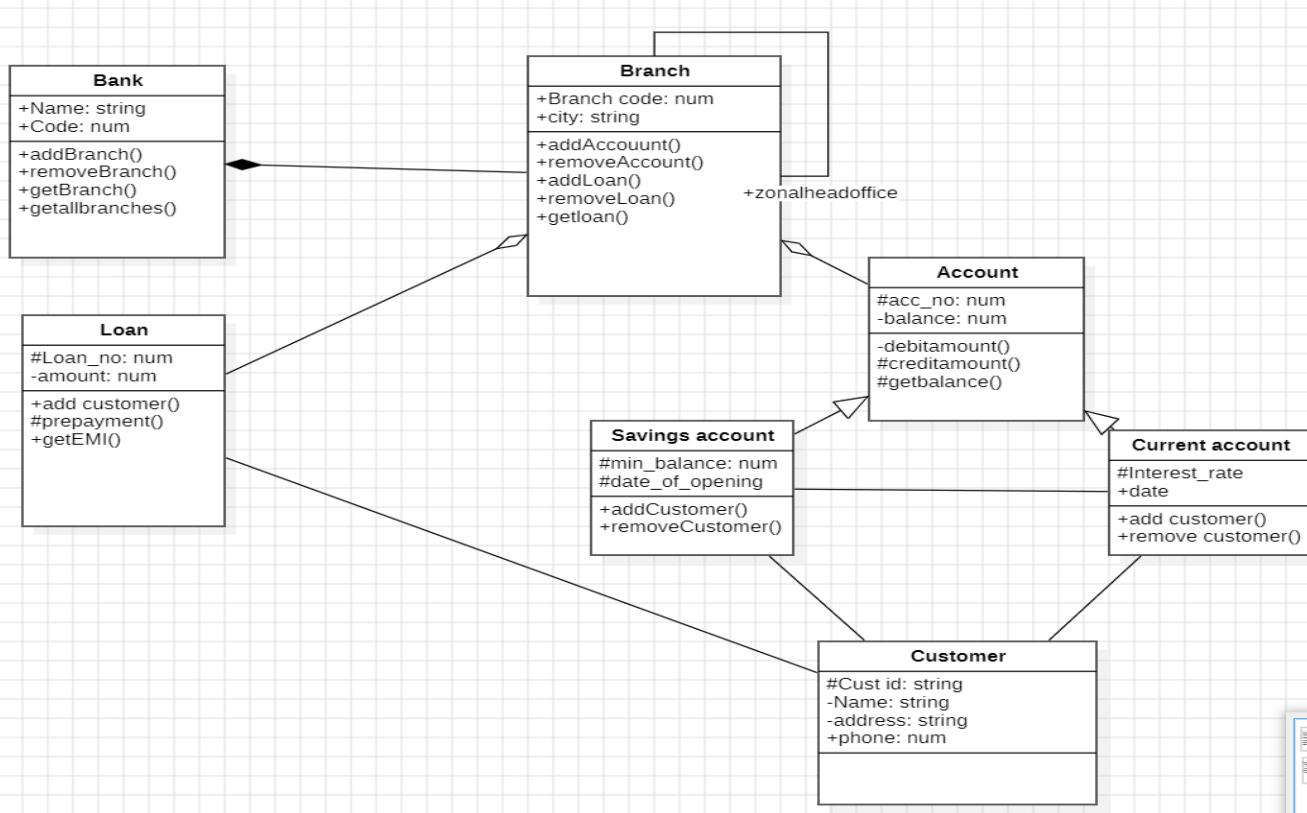
UML DIAGRAMS

1) BANK MANAGEMENT SYSTEM

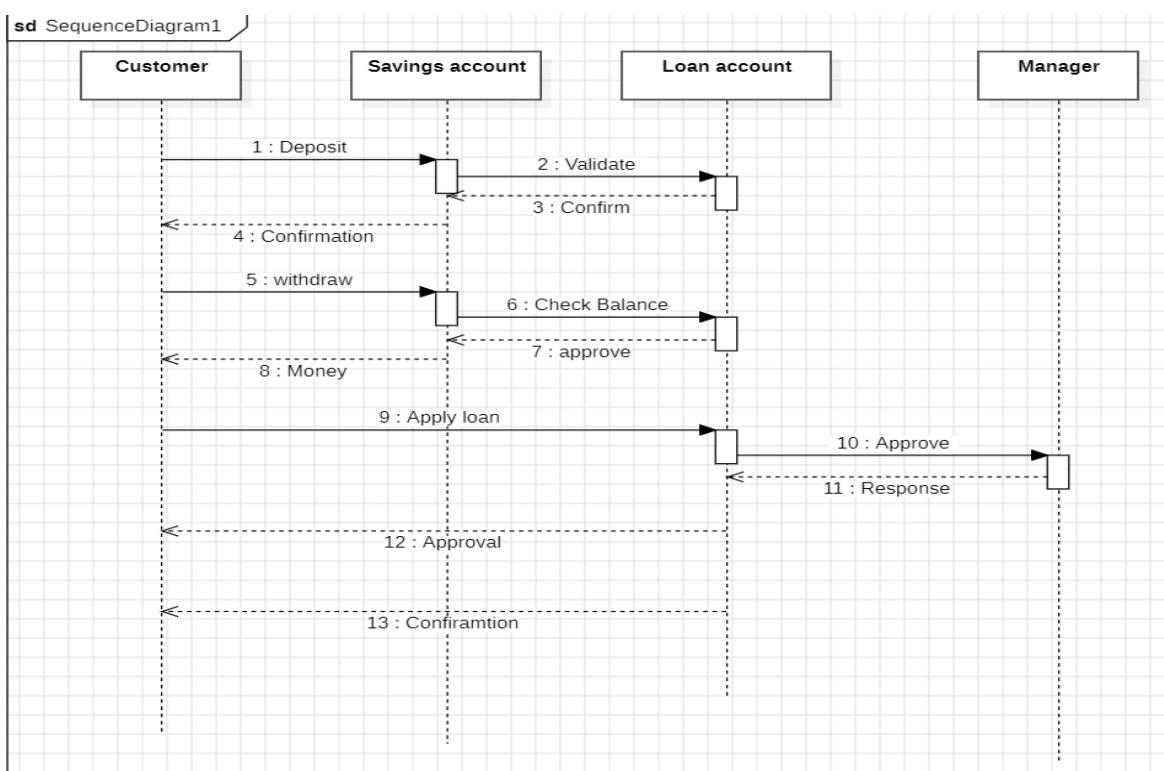
1.a) Use case diagram



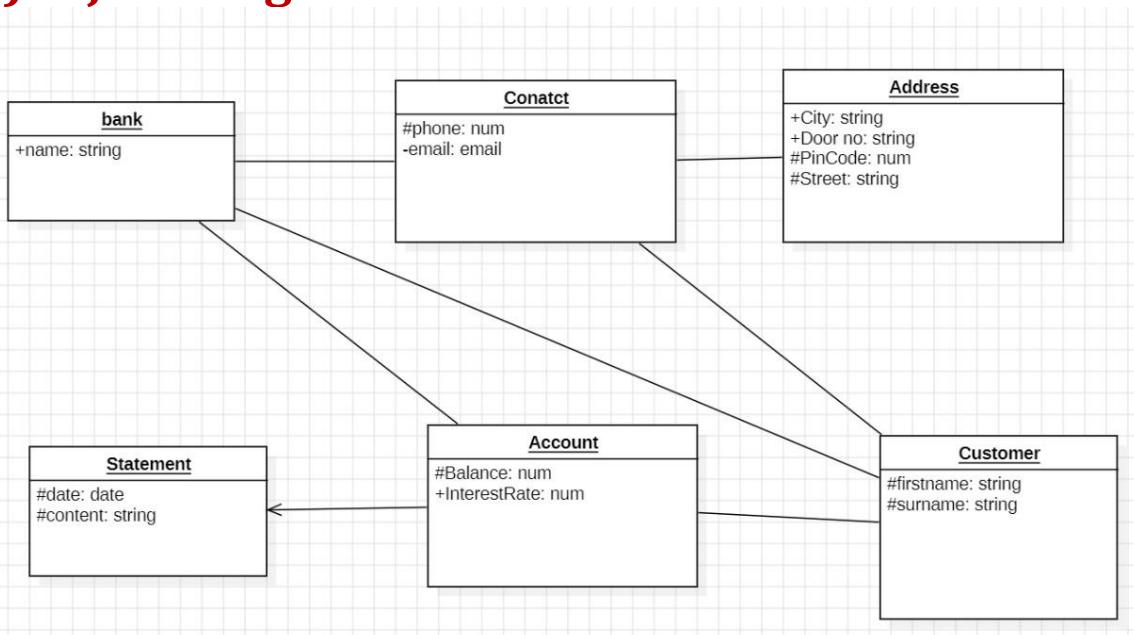
1b)class diagram



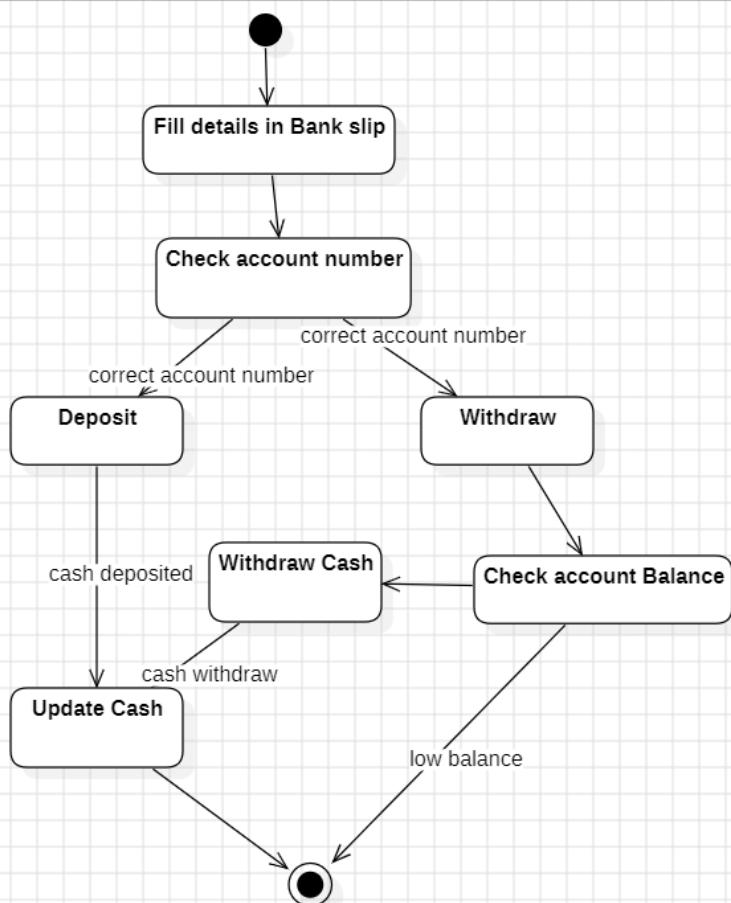
1c)sequence diagram



1d) object diagram

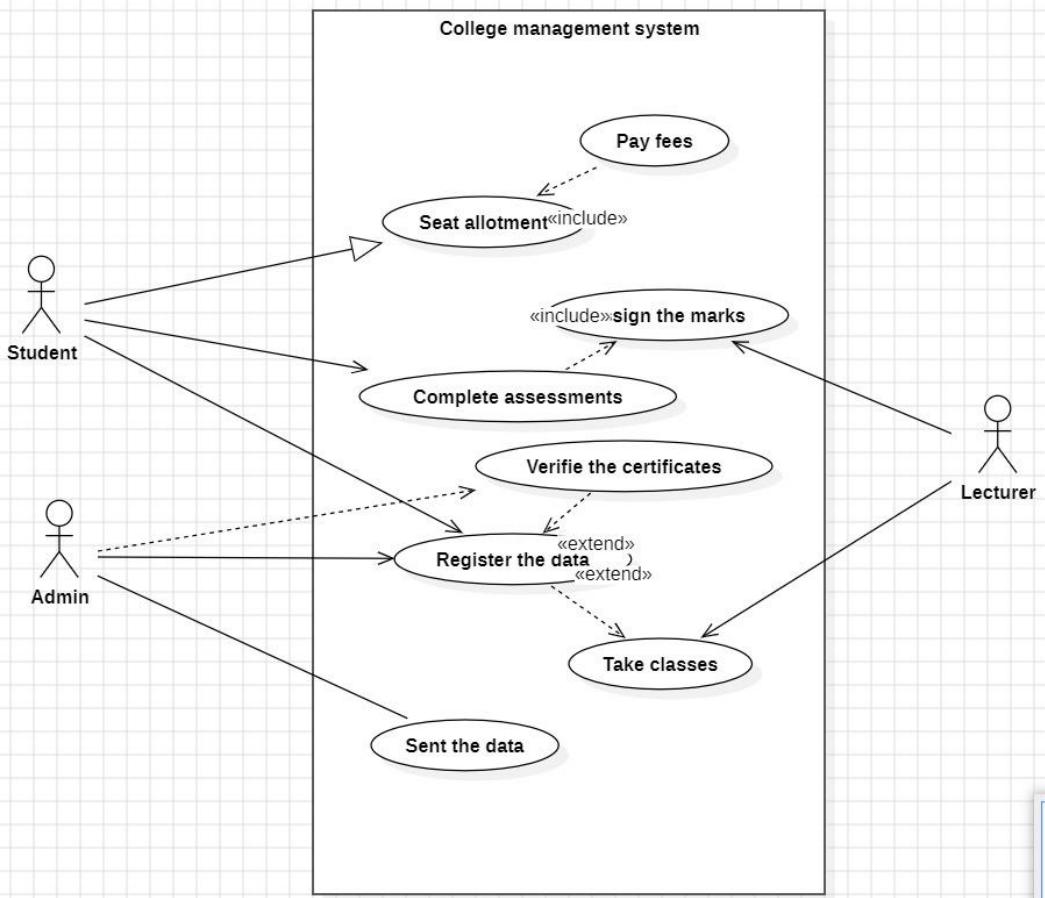


1e) state diagram



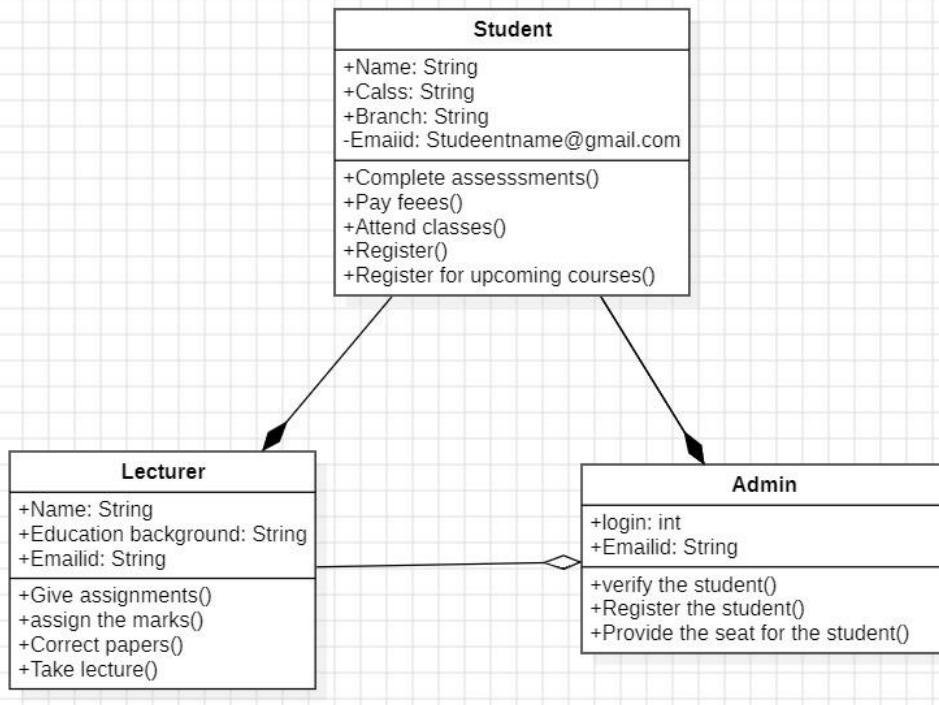
2.COLLEGE MANAGEMENT SYSTEM

2a)usecase diagram

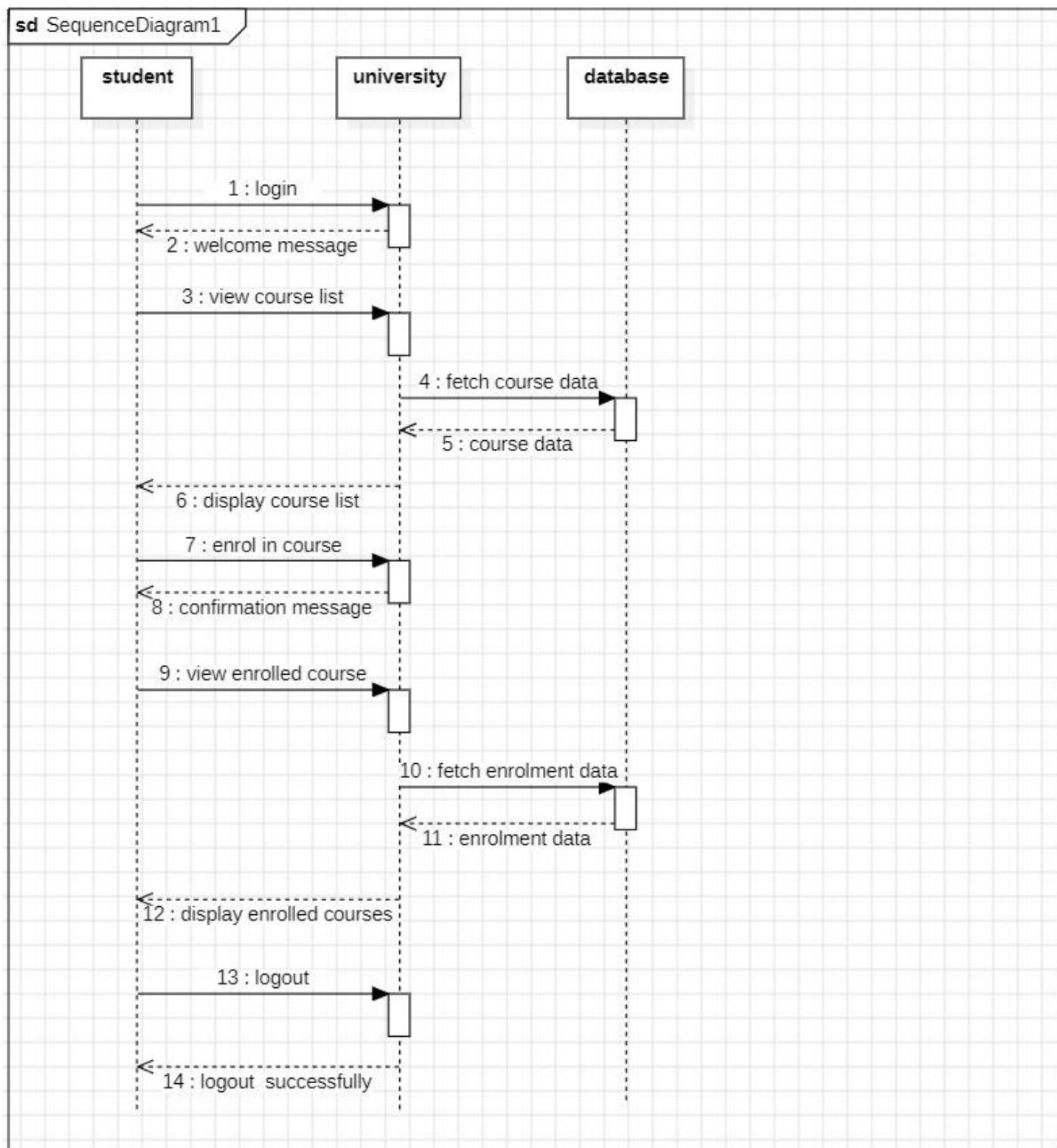


2b)class diagram

COLLEGE MANAGEMENT SYSTEM

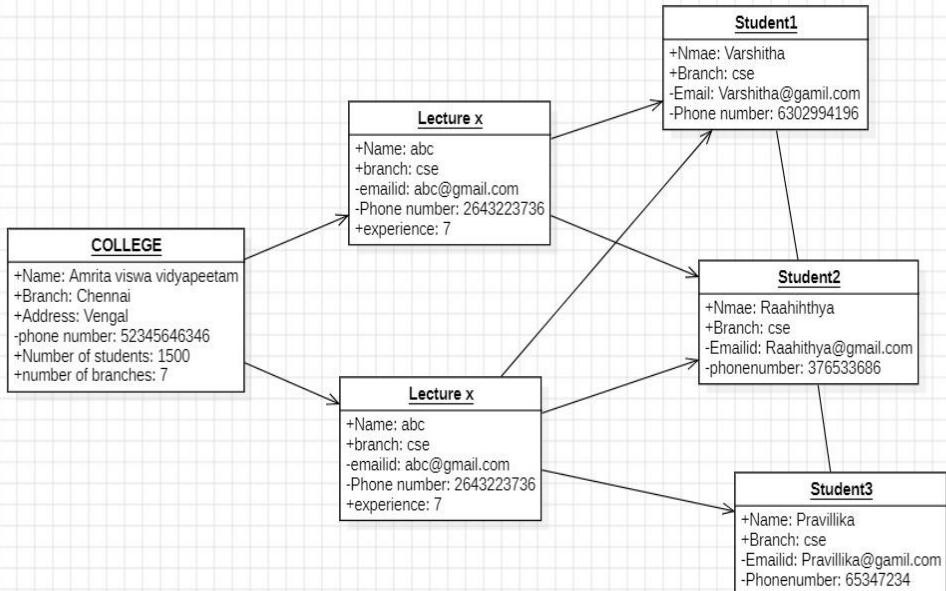


2c)sequence diagram

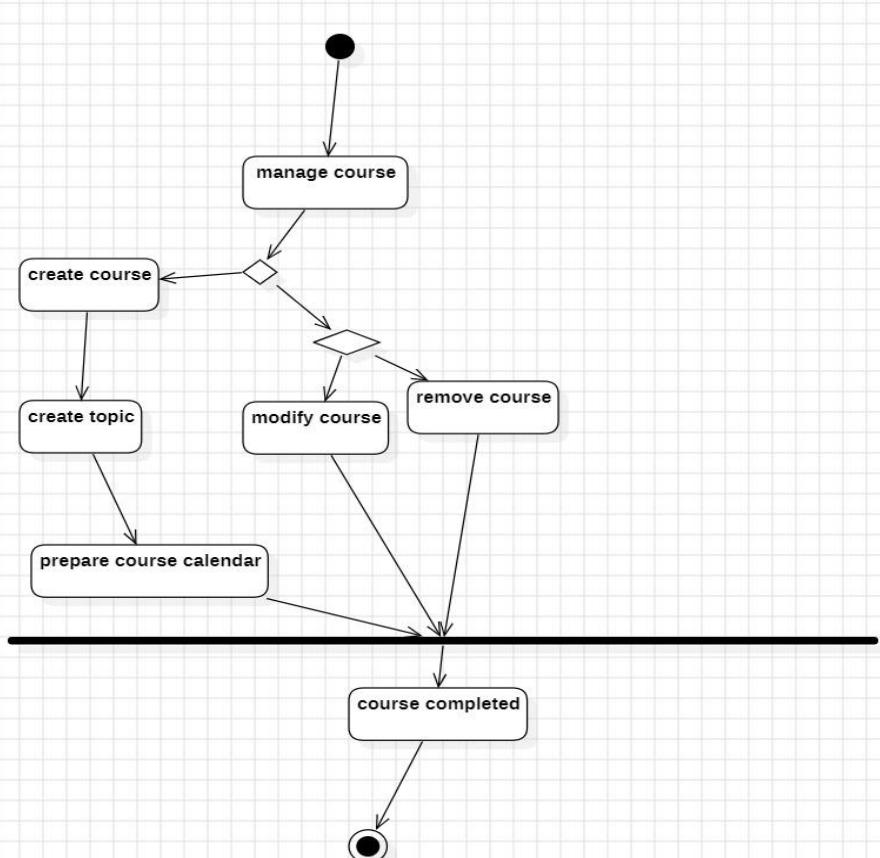


2d) object diagram

COLLEGE MANAGEMENT SYSTEM



2e) state diagram



3.Basic Java Programs

a) AIM: To write a Program to display details of a student by input
CODE:

```
1 import java.util.Scanner;
2 public class Student {
3     String name;
4     int age;
5     public void displayInfo() {
6         String status = (age >= 18) ? "Adult" : "Minor";
7         System.out.println("Name: " + name);
8         System.out.println("Age: " + age);
9         System.out.println("Status: " + status);
10    }
11
12 public static void main(String[] args) {
13 Scanner scanner = new Scanner(System.in);
14 Student student1 = new Student();
15 System.out.print("Enter first student's name: ");
16 student1.name = scanner.nextLine();
17 System.out.print("Enter first student's age: ");
18 student1.age = scanner.nextInt();
19 scanner.nextLine();
20
21 student1.displayInfo();
22
23 scanner.close();
24    }
25 }
```

OUTPUT:

```
Enter first student's name: Raahithya
Enter first student's age: 19
Name: Raahithya
Age: 19
Status: Adult
```

b) AIM: To Print Pattern by taking input from user.

CODE:

```
1 import java.util.Scanner;
2 public class PrintPattern {
3     public static void main(String[] args) {
4         Scanner scanner = new Scanner(System.in);
5         System.out.print("Enter the number of rows: ");
6         int rows = scanner.nextInt();
7
8         for (int i = 1; i <= rows; ++i) {
9             for (int j = 1; j <= i; ++j) {
10                 System.out.print("* ");
11             }
12             System.out.println();
13         }
14         scanner.close();
15     }
16 }
17
```

OUTPUT:

c)AIM: To check if the given letter is vowel or not

CODE:

```
1 import java.util.Scanner;
2 public class VowelConsonent {
3     public static void main(String[] args) {
4         Scanner scanner = new Scanner(System.in);
5         System.out.print("Enter a character: ");
6         char ch = scanner.next().charAt(0);
7         if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
8             ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {
9             System.out.println(ch + " is a vowel.");
10        } else if (Character.isLetter(ch)) {
11            System.out.println(ch + " is a consonant.");
12        } else {
13            System.out.println(ch + " is not a valid letter.");
14        }
15    }
16    scanner.close();
17 }
18 }
19 }
```

OUTPUT:

```
Enter a character: g
g is a consonant.
```

d)AIM: To find the Sum of digits of a number

CODE:

```
1 import java.util.Scanner;
2 public class SumOfDigits {
3     public static void main(String[] args) {
4         Scanner scanner = new Scanner(System.in);
5         System.out.print("Enter a number: ");
6         int num = scanner.nextInt();
7
8         int sum = 0;
9
10        int originalNum = num;
11
12        while (num > 0) {
13            int digit = num % 10;
14            sum += digit;
15            num /= 10;
16        }
17        System.out.println("Sum of digits of " + originalNum + " is: " + sum);
18
19        scanner.close();
20    }
21 }
22 }
```

OUTPUT:

```
Enter a number: 7649
Sum of digits of 7649 is: 26
```

e)AIM: To reverse a number**CODE:**

```
1 import java.util.Scanner;
2 public class ReverseNumber {
3     public static void main(String[] args) {
4         Scanner scanner = new Scanner(System.in);
5
6         System.out.print("Enter a number: ");
7         int num = scanner.nextInt();
8
9         int reverse = 0;
10
11        for (; num > 0; num /= 10) {
12            int digit = num % 10;
13            reverse = reverse * 10 + digit;
14        }
15        System.out.println("Reversed Number: " + reverse);
16
17        scanner.close();
18    }
19
20 }
```

Output:

```
Enter a number: 9087
Reversed Number: 7809
```

f) AIM: To find Fibonacci series.

CODE:

```
1 import java.util.Scanner;
2 public class Fibonacci {
3     public static void main(String[] args) {
4         Scanner scanner = new Scanner(System.in);
5         System.out.print("Enter the number of terms: ");
6         int n = scanner.nextInt();
7         int first = 0, second = 1;
8         System.out.println("Fibonacci Series up to " + n + " terms:");
9
10        for (int i = 1; i <= n; i++) {
11            System.out.print(first + " ");
12            int next = first + second;
13            first = second;
14            second = next;
15        }
16
17        scanner.close();
18    }
19 }
20
```

OUTPUT:

```
Enter the number of terms: 9
Fibonacci Series up to 9 terms:
0 1 1 2 3 5 8 13 21
```

g)AIM: Deposit and Withdrawal from a Bank Account

CODE:

```
1 class BankAccount {  
2     String accountHolderName;  
3     double balance;  
4     BankAccount(String name, double initialBalance) {  
5         accountHolderName = name;  
6         balance = initialBalance;  
7     }  
8  
9     void deposit(double amount) {  
10        balance += amount;  
11        System.out.println("Deposited: " + amount + ", New Balance: " + balance);  
12    }  
13  
14    void withdraw(double amount) {  
15        if (amount > balance) {  
16            System.out.println("Insufficient balance!");  
17        } else {  
18            balance -= amount;  
19            System.out.println("Withdrawn: " + amount + ", New Balance: " + balance);  
20        }  
21    }  
22  
23    public static void main(String[] args) {  
24        BankAccount account = new BankAccount("Alice", 1000);  
25        account.deposit(500);  
26        account.withdraw(300);  
27        account.withdraw(1500);  
28    }  
29 }  
30 }
```

OUTPUT:

```
Deposited: 500.0, New Balance: 1500.0  
Withdrawn: 300.0, New Balance: 1200.0  
Insufficient balance!
```

h)AIM: To display grade of a student.

CODE:

```
1 import java.util.Scanner;
2
3 public class StudentGrade {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.print("Enter the student's marks (0-100): ");
7         int marks = scanner.nextInt();
8         if (marks >= 90) {
9             System.out.println("Grade: A+ (Excellent)");
10        } else if (marks >= 80) {
11            System.out.println("Grade: A (Very Good)");
12        } else if (marks >= 70) {
13            System.out.println("Grade: B (Good)");
14        } else if (marks >= 60) {
15            System.out.println("Grade: C (Satisfactory)");
16        } else if (marks >= 50) {
17            System.out.println("Grade: D (Pass)");
18        } else {
19            System.out.println("Grade: F (Fail)");
20        }
21
22        scanner.close();
23    }
24 }
```

OUTPUT:

```
Enter the student's marks (0-100): 76
Grade: B (Good)
```

i) AIM: To find area of a circle.

CODE:

```
1 import java.util.Scanner;
2 public class CircleArea {
3     public static void main(String[] args) {
4
5         Scanner scanner = new Scanner(System.in);
6
7         System.out.print("Enter the radius of the circle: ");
8         double radius = scanner.nextDouble();
9         double area = Math.PI * radius * radius;
10
11        System.out.println("The area of the circle is: " + area);
12
13        scanner.close();
14    }
15 }
16
```

OUTPUT:

```
Enter the radius of the circle: 9
The area of the circle is: 254.46900494077323
```

J) AIM: To write a program for simple calculator

CODE:

```

1 import java.util.Scanner;
2
3 public class Calculator {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         System.out.print("Enter first number: ");
8         double num1 = scanner.nextDouble();
9
10        System.out.print("Enter second number: ");
11        double num2 = scanner.nextDouble();
12
13        System.out.print("Choose an operation (+, -, *, /): ");
14        char operator = scanner.next().charAt(0);
15
16        double result;
17
18        switch (operator) {
19            case '+':
20                result = num1 + num2;
21                System.out.println("Result: " + result);
22                break;
23            case '-':
24                result = num1 - num2;
25                System.out.println("Result: " + result);
26                break;
27            case '*':
28                result = num1 * num2;
29                System.out.println("Result: " + result);
30                break;
31            case '/':
32                if (num2 != 0) {
33                    result = num1 / num2;
34                    System.out.println("Result: " + result);
35                } else {
36                    System.out.println("Error! Division by zero is not allowed.");
37                }
38                break;
39            default:
40                System.out.println("Invalid operator! Please choose +, -, *, or /.");
41        }
42
43        scanner.close();
44    }
45 }
46

```

OUTPUT:

```

Enter first number: 5
Enter second number: 8
Choose an operation (+, -, *, /): /
Result: 0.625

```

Inheritance

4)Single Inheritance

4.a)

Code:

```
1 class Appliance {  
2     public void turnOn() {  
3         System.out.println("Appliance is now ON.");  
4     }  
5     public void turnOff() {  
6         System.out.println("Appliance is now OFF.");  
7     }  
8 }  
9 class WashingMachine extends Appliance {  
10    void washClothes() {  
11        System.out.println("Washing machine is washing clothes");  
12    }  
13 }  
14 public class Electronics {  
15     public static void main(String[] args) {  
16         WashingMachine myWasher = new WashingMachine();  
17         myWasher.turnOn();  
18         myWasher.turnOff();  
19         myWasher.washClothes();  
20     }  
21 }  
22 }
```

Output:

```
Appliance is now ON.  
Appliance is now OFF.  
Washing machine is washing clothes
```

4.b)Code:

```

1 import java.util.Scanner;
2 class Employee {
3     String name;
4     double salary;
5     void getEmployeeDetails() {
6         Scanner sc = new Scanner(System.in);
7         System.out.print("Enter Employee Name: ");
8         name = sc.nextLine();
9         System.out.print("Enter Salary: ");
10        salary = sc.nextDouble();
11    }
12    void displayEmployeeDetails() {
13        System.out.println("Employee Name: " + name);
14        System.out.println("Salary:" + salary);
15    }
16 }
17 class Manager extends Employee {
18     String department;
19     void getManagerDetails() {
20         Scanner sc = new Scanner(System.in);
21         System.out.print("Enter Department: ");
22         department = sc.nextLine();
23     }
24     void displayManagerDetails() {
25         displayEmployeeDetails();
26         System.out.println("Department: " + department);
27     }
28 }
29 public class Inheritance {
30     public static void main(String[] args) {
31         Manager manager = new Manager();
32         manager.getEmployeeDetails();
33         manager.getManagerDetails();
34         System.out.println("\nManager Details:");
35         manager.displayManagerDetails();      }
36 }
```

Output:

```

Enter Employee Name: Raahithya
Enter Salary: 100000
Enter Department: execution
```

```

Manager Details:
Employee Name: Raahithya
Salary:100000.0
Department: execution
```

5) MULTILEVEL INHERITANCE PROGRAMS

5.a) Code:

```

1 import java.util.Scanner;
2
3 class Number {
4     int n;
5
6     void getInput() {
7         Scanner sc = new Scanner(System.in);
8         System.out.print("Enter the number of Fibonacci terms: ");
9         n = sc.nextInt();
10        sc.close();
11    }
12 }
13
14 class Fibonacci extends Number {
15     int[] fibonacciSeries;
16
17     void generateFibonacci() {
18         fibonacciSeries = new int[n];
19         fibonacciSeries[0] = 0;
20         if (n > 1) {
21             fibonacciSeries[1] = 1;
22             for (int i = 2; i < n; i++) {
23                 fibonacciSeries[i] = fibonacciSeries[i - 1] + fibonacciSeries[i - 2];
24             }
25         }
26     }
27 }
28
29 class FibonacciSum extends Fibonacci {
30     int sum = 0;
31
32     void calculateSum() {
33         for (int num : fibonacciSeries) {
34             sum += num;
35         }
36     }
37
38     void display() {
39         System.out.println("Fibonacci Series: ");
40         for (int num : fibonacciSeries) {
41             System.out.println(num + " ");
42         }
43         System.out.println("Sum of Fibonacci Series: " + sum);
44     }
45 }
46
47 public class FibonacciInheritance {
48     public static void main(String[] args) {
49         FibonacciSum obj = new FibonacciSum();
50         obj.getInput();
51         obj.generateFibonacci();
52         obj.calculateSum();
53         obj.display();
54     }
55 }
56

```

Output:

```
Enter the number of Fibonacci terms: 15
Fibonacci Series:
0
1
1
2
3
5
8
13
21
34
55
89
144
233
377
Sum of Fibonacci Series: 986
```

5.b) Code:

```

1 import java.util.Random;
2 import java.util.Scanner;
3
4 class Game {
5     int rolls;
6
7     void getInput() {
8         Scanner sc = new Scanner(System.in);
9         System.out.print("Enter the number of times you want to roll the dice: ");
10        rolls = sc.nextInt();
11        sc.close();
12    }
13 }
14
15 class Dice extends Game {
16     int[] outcomes;
17
18     void rollDice() {
19         Random rand = new Random();
20         outcomes = new int[rolls];
21         for (int i = 0; i < rolls; i++) {
22             outcomes[i] = rand.nextInt(6) + 1;
23         }
24     }
25 }
26
27 class Probability extends Dice {
28     double probabilityOfSix;
29
30     void calculateProbability() {
31         int countSix = 0;
32         for (int outcome : outcomes) {
33             if (outcome == 6) {
34                 countSix++;
35             }
36         }
37         probabilityOfSix = (double) countSix / rolls;
38     }
39
40     void display() {
41         System.out.print("Dice Rolls: ");
42         for (int num : outcomes) {
43             System.out.print(num + " ");
44         }
45         System.out.println("\nTimes 6 appeared: " + (int) (probabilityOfSix * rolls));
46         System.out.println("Probability of rolling a 6: " + probabilityOfSix);
47     }
48 }
49
50 public class DiceProbability {
51     public static void main(String[] args) {
52         Probability obj = new Probability();
53         obj.getInput();
54         obj.rollDice();
55         obj.calculateProbability();
56         obj.display();
57     }
58 }
59

```

Output:

```
Enter the number of times you want to roll the dice: 4
Dice Rolls: 5 2 5 3
Times 6 appeared: 0
Probability of rolling a 6: 0.0
```

6) HIRARCHIAL INHERITANCE PROGRAMS

6.a) Code:

```

1 class Planet {
2     String name;
3     double mass;
4     double radius;
5     double gravity;
6
7     void display() {
8         System.out.println("Planet: " + name);
9         System.out.println("Mass: " + mass + " kg");
10        System.out.println("Radius: " + radius + " km");
11        System.out.println("Gravity: " + gravity + " m/s2");
12        System.out.println("-----");
13    }
14 }
15
16 class Earth extends Planet {
17     Earth() {
18         name = "Earth";
19         mass = 5.972 * Math.pow(10, 24);
20         radius = 6371;
21         gravity = 9.8;
22     }
23 }
24
25 class Mars extends Planet {
26     Mars() {
27         name = "Mars";
28         mass = 6.39 * Math.pow(10, 23);
29         radius = 3389.5;
30         gravity = 3.71;
31     }
32 }
33
34 class Venus extends Planet {
35     Venus() {
36         name = "Venus";
37         mass = 4.867 * Math.pow(10, 24);
38         radius = 6051.8;
39         gravity = 8.87;
40     }
41 }
42
43 public class PlanetDetails {
44     public static void main(String[] args) {
45         Earth e = new Earth();
46         Mars m = new Mars();
47         Venus v = new Venus();
48
49         e.display();
50         m.display();
51         v.display();
52     }
53 }
```

Output:

```
Planet: Earth
Mass: 5.972E24 kg
Radius: 6371.0 km
Gravity: 9.8 m/s2
-----
```

```
Planet: Mars
Mass: 6.39E23 kg
Radius: 3389.5 km
Gravity: 3.71 m/s2
-----
```

```
Planet: Venus
Mass: 4.867E24 kg
Radius: 6051.8 km
Gravity: 8.87 m/s2
-----
```

6.b)

Code:

```
1 class Time {  
2     int hours = 24;  
3 }  
4  
5 class Minute extends Time {  
6     int toMinutes() {  
7         return hours * 60;  
8     }  
9 }  
10  
11 class Second extends Time {  
12     int toSeconds() {  
13         return hours * 3600;  
14     }  
15 }  
16  
17 public class TimeConverter {  
18     public static void main(String[] args) {  
19         Minute min = new Minute();  
20         Second sec = new Second();  
21  
22         System.out.println("Hours: " + min.hours);  
23         System.out.println("Minutes: " + min.toMinutes());  
24         System.out.println("Seconds: " + sec.toSeconds());  
25     }  
26 }  
27
```

Output:

```
Hours: 24  
Minutes: 1440  
Seconds: 86400
```

7)HYBRID INHERITANCE PROGRAM

7.a)Code:

```
1 class Country {  
2     void countryInfo() {  
3         System.out.println("This is a country.");  
4     }  
5 }  
6  
7 class State extends Country {  
8     void stateInfo() {  
9         System.out.println("This is a state within a country.");  
10    }  
11 }  
12  
13 class District extends State {  
14     void districtInfo() {  
15         System.out.println("This is a district within a state.");  
16     }  
17 }  
18  
19 class City extends State {  
20     void cityInfo() {  
21         System.out.println("This is a city within a state.");  
22     }  
23 }  
24  
25 public class HybridCountry {  
26     public static void main(String[] args) {  
27         District d = new District();  
28         City c = new City();  
29  
30         d.countryInfo();  
31         d.stateInfo();  
32         d.districtInfo();  
33         System.out.println("-----");  
34         c.countryInfo();  
35         c.stateInfo();  
36         c.cityInfo();  
37     }  
38 }  
39
```

Output:

```
This is a country.  
This is a state within a country.  
This is a district within a state.  
-----  
This is a country.  
This is a state within a country.  
This is a city within a state.
```

7.b)Code:

```
1 class ShoppingCart {  
2     void cartInfo() {  
3         System.out.println("This is a Shopping Cart.");  
4     }  
5 }  
6  
7 class Payment extends ShoppingCart {  
8     void paymentInfo() {  
9         System.out.println("Payment methods are available.");  
10    }  
11 }  
12  
13 class UPI extends Payment {  
14     void upiPayment() {  
15         System.out.println("Payment done via UPI.");  
16     }  
17 }  
18  
19 class CreditCard extends Payment {  
20     void creditCardPayment() {  
21         System.out.println("Payment done via Credit Card.");  
22     }  
23 }  
24  
25 public class Shopping {  
26     public static void main(String[] args) {  
27         UPI upi = new UPI();  
28         CreditCard cc = new CreditCard();  
29  
30         upi.cartInfo();  
31         upi.paymentInfo();  
32         upi.upiPayment();  
33         System.out.println("-----");  
34         cc.cartInfo();  
35         cc.paymentInfo();  
36         cc.creditCardPayment();  
37     }  
38 }  
39
```

Output:

```
This is a Shopping Cart.  
Payment methods are available.  
Payment done via UPI.  
-----
```

```
This is a Shopping Cart.  
Payment methods are available.  
Payment done via Credit Card.
```

POLYMORPHISM PROGRAMS

8)CONSTRUCTOR PROGRAM

8.a)

Code:

```
1 class Stock {  
2     String companyName;  
3     double currentPrice;  
4     double changePercentage;  
5  
6     Stock(String name, double price, double change) {  
7         companyName = name;  
8         currentPrice = price;  
9         changePercentage = change;  
10    }  
11  
12    void showStockDetails() {  
13        System.out.println("Company: " + companyName);  
14        System.out.println("Current Price:" + currentPrice+/-");  
15        System.out.println("Price Change: " + changePercentage + "%");  
16        if (changePercentage > 0) {  
17            System.out.println("Status: Gained");  
18        } else if (changePercentage < 0) {  
19            System.out.println("Status: Lost");  
20        } else {  
21            System.out.println("Status: No Change");  
22        }  
23    }  
24  
25    public static void main(String[] args) {  
26        Stock s1 = new Stock("Tata Motors", 860.50, 1.75);  
27        Stock s2 = new Stock("Infosys", 1400.25, -0.50);  
28  
29        s1.showStockDetails();  
30  
31        s2.showStockDetails();  
32    }  
33 }  
34 }
```

Output:

Company: Tata Motors

Current Price: 860.5/-

Price Change: 1.75%

Status: Gained

Company: Infosys

Current Price: 1400.25/-

Price Change: -0.5%

Status: Lost

9)CONSTRUCTOR OVERLOADING PROGRAM

9.a)Code:

```

1 import java.util.Scanner;
2
3 class Player {
4     String name;
5     String sport;
6     int matches;
7     int totalScore;
8
9     Player() {
10         name = "Unknown";
11         sport = "Unknown";
12         matches = 0;
13         totalScore = 0;
14     }
15
16     Player(String playerName, String playerSport) {
17         name = playerName;
18         sport = playerSport;
19         matches = 0;
20         totalScore = 0;
21     }
22
23     Player(String playerName, String playerSport, int games, int score) {
24         name = playerName;
25         sport = playerSport;
26         matches = games;
27         totalScore = score;
28     }
29
30     void showPerformance() {
31         double average = (matches > 0) ? (double) totalScore / matches : 0;
32         System.out.println("\nPlayer: " + name);
33         System.out.println("Sport: " + sport);
34         System.out.println("Matches Played: " + matches);
35         System.out.println("Total Score: " + totalScore);
36         System.out.println("Average Score per Match: " + average);
37     }
38
39     public static void main(String[] args) {
40         Scanner sc = new Scanner(System.in);
41
42         System.out.print("Enter Player Name: ");
43         String playerName = sc.nextLine();
44
45         System.out.print("Enter Sport Name: ");
46         String playerSport = sc.nextLine();
47
48         System.out.print("Enter Matches Played: ");
49         int matches = sc.nextInt();
50
51         System.out.print("Enter Total Score: ");
52         int totalScore = sc.nextInt();
53
54         Player userPlayer = new Player(playerName, playerSport, matches, totalScore);
55
56         userPlayer.showPerformance();
57
58
59         sc.close();
60     }
61 }
```

Output:

```
Enter Player Name: Virat Kohli
Enter Sport Name: Cricket
Enter Matches Played: 900
Enter Total Score: 12000

Player: Virat Kohli
Sport: Cricket
Matches Played: 900
Total Score: 12000
Average Score per Match: 13.33333333333334
```

10)METHOD OVERLOADING PROGRAMS

10.a)

Code:

```
1 class PowerCalculator {  
2     int power(int x) {  
3         return x * x;  
4     }  
5  
6     int power(int x, int exponent) {  
7         return x * x * x;  
8     }  
9  
10    public static void main(String[] args) {  
11        PowerCalculator obj = new PowerCalculator();  
12        System.out.println("Square of 5: " + obj.power(9));  
13        System.out.println("Cube of 3: " + obj.power(5, 7));  
14    }  
15}  
16
```

Output:

```
Square of 5: 81  
Cube of 3: 125
```

10.b)

Code:

```
1 import java.util.Scanner;
2
3 class AgeCalculator {
4     int calculateAge(int birthYear, int currentYear) {
5         return currentYear - birthYear;
6     }
7
8     int calculateAge(int birthYear, int currentYear, int birthMonth, int currentMonth) {
9         int years = currentYear - birthYear;
10        int months = (years * 12) + (currentMonth - birthMonth);
11        return months;
12    }
13
14    public static void main(String[] args) {
15        Scanner sc = new Scanner(System.in);
16
17        System.out.print("Enter birth year: ");
18        int birthYear = sc.nextInt();
19
20        System.out.print("Enter birth month (1-12): ");
21        int birthMonth = sc.nextInt();
22
23        System.out.print("Enter current year: ");
24        int currentYear = sc.nextInt();
25
26        System.out.print("Enter current month (1-12): ");
27        int currentMonth = sc.nextInt();
28
29        AgeCalculator obj = new AgeCalculator();
30        int ageInYears = obj.calculateAge(birthYear, currentYear);
31        int ageInMonths = obj.calculateAge(birthYear, currentYear, birthMonth, currentMonth);
32
33        System.out.println("\nYour age in years: " + ageInYears);
34        System.out.println("Your age in months: " + ageInMonths);
35
36        sc.close();
37    }
38}
39
```

Output:

```
Enter birth year: 1979
Enter birth month (1-12): 06
Enter current year: 2025
Enter current month (1-12): 04

Your age in years: 46
Your age in months: 550
```

11)METHOD OVERRIDING PROGRAMS

11.a)

Code:

```

1 class Factorial {
2     int calculate(int n) {
3         int fact = 1;
4         for (int i = 1; i <= n; i++) {
5             fact *= i;
6         }
7         return fact;
8     }
9 }
10
11class RecursiveFactorial extends Factorial {
12     int calculate(int n) {
13         if (n == 0 || n == 1) {
14             return 1;
15         }
16         return n * calculate(n - 1);
17     }
18
19     public static void main(String[] args) {
20         Factorial iterative = new Factorial();
21         RecursiveFactorial recursive = new RecursiveFactorial();
22
23         int num = 9;
24         System.out.println("Factorial of " + num + " (Iterative): " + iterative.calculate(num));
25         System.out.println("Factorial of " + num + " (Recursive): " + recursive.calculate(num));
26     }
27}
28

```

Output:

```

Factorial of 9 (Iterative): 362880
Factorial of 9 (Recursive): 362880

```

11.b)

Code:

```
1 class SecuritySystem {
2     void authenticate() {
3         System.out.println("Authenticating using basic password...");
4     }
5 }
6
7 class FingerprintScanner extends SecuritySystem {
8     void authenticate() {
9         System.out.println("Authenticating using Fingerprint Scanner...");
10    }
11}
12
13class FaceRecognition extends SecuritySystem {
14    void authenticate() {
15        System.out.println("Authenticating using Face Recognition...");
16    }
17}
18
19public class Secure {
20    public static void main(String[] args) {
21        SecuritySystem basic = new SecuritySystem();
22        SecuritySystem fingerprint = new FingerprintScanner();
23        SecuritySystem face = new FaceRecognition();
24
25        basic.authenticate();
26        fingerprint.authenticate();
27        face.authenticate();
28    }
29}
30
```

Output:

```
Authenticating using basic password...
Authenticating using Fingerprint Scanner...
Authenticating using Face Recognition...
```

ABSTRACTION PROGRAMS

12)INTERFACE PROGRAMS

12.a)

Code:

```
1 interface ATMabs {
2     void withdraw(double amount);
3     void deposit(double amount);
4 }
5
6 class BankATM implements ATMabs {
7     private double balance = 200000;
8
9     public void withdraw(double amount) {
10         if (amount > balance) {
11             System.out.println("Insufficient Balance!");
12         } else {
13             balance -= amount;
14             System.out.println("Withdrawn: " + amount + ", Remaining Balance: " + balance);
15         }
16     }
17
18     public void deposit(double amount) {
19         balance += amount;
20         System.out.println("Deposited: " + amount + ", Total Balance: " + balance);
21     }
22 }
23
24 public class ATM {
25     public static void main(String[] args) {
26         ATMabs atm = new BankATM();
27         atm.deposit(20000);
28         atm.withdraw(7000);
29     }
30 }
```

Output:

```
Deposited: 20000.0, Total Balance: 220000.0
Withdrawn: 7000.0, Remaining Balance: 213000.0
```

12.b)

Code:

```
1 interface Polygon {  
2     void getArea();  
3  
4     default void getSides() {  
5         System.out.println("I can get sides of a polygon.");  
6     }  
7 }  
8  
9 class Rectangle implements Polygon {  
10    public void getArea() {  
11        int length = 9;  
12        int breadth = 5;  
13        int area = length * breadth;  
14        System.out.println("The area of the rectangle is " + area);  
15    }  
16  
17    public void getSides() {  
18        System.out.println("It has 4 sides.");  
19    }  
20 }  
21  
22 class Square implements Polygon {  
23    public void getArea() {  
24        int length = 6;  
25        int area = length * length;  
26        System.out.println("The area of the square is " + area);  
27    }  
28 }  
29  
30 class Shapes {  
31    public static void main(String[] args) {  
32  
33        Rectangle r1 = new Rectangle();  
34        r1.getArea();  
35        r1.getSides();  
36  
37        Square s1 = new Square();  
38        s1.getArea();  
39        s1.getSides();  
40    }  
41 }
```

Output:

The area of the rectangle is 45

It has 4 sides.

The area of the square is 36

I can get sides of a polygon.

12.c)

Code:

```

1 interface Payment {
2     void processPayment(double amount);
3
4     default void printReceipt(double amount) {
5         System.out.println("Receipt: Payment of " + amount + "/- completed.");
6     }
7 }
8
9 class CreditCardPayment implements Payment {
10    public void processPayment(double amount) {
11        System.out.println("Processing Credit Card Payment of " + amount+"/-");
12        printReceipt(amount);
13    }
14 }
15
16 class UPIPayment implements Payment {
17    public void processPayment(double amount) {
18        System.out.println("Processing UPI Payment of " + amount+"/-");
19        printReceipt(amount);
20    }
21 }
22
23 public class Pay {
24     public static void main(String[] args) {
25         Payment credit = new CreditCardPayment();
26         credit.processPayment(5000);
27
28         Payment upi = new UPIPayment();
29         upi.processPayment(1500);
30     }
31 }
32

```

Output:

```

Processing Credit Card Payment of 5000.0/-
Receipt: Payment of 5000.0/- completed.
Processing UPI Payment of 1500.0/-
Receipt: Payment of 1500.0/- completed.

```

12.d)Code:

```

1 import java.util.Scanner;
2 interface MathOperations {
3     int factorial(int num);
4     int gcd(int a, int b);
5     boolean isPrime(int num);}
6 interface AdvancedMathOperations {
7     void fibonacci(int terms);
8     double power(double base, int exponent);}
9 class BasicMath implements MathOperations {
10    public int factorial(int num) {
11        if (num == 0 || num == 1) return 1;
12        return num * factorial(num - 1); }
13    public int gcd(int a, int b) {
14        if (b == 0) return a;
15        return gcd(b, a % b); }
16    public boolean isPrime(int num) {
17        if (num < 2) return false;
18        for (int i = 2; i <= Math.sqrt(num); i++) {
19            if (num % i == 0) return false;   }
20        return true; }}
21 class AdvancedMath implements AdvancedMathOperations {
22    public void fibonacci(int terms) {
23        int a = 0, b = 1;
24        System.out.print("Fibonacci Series: ");
25        for (int i = 0; i < terms; i++) {
26            System.out.print(a + " ");
27            int next = a + b;
28            a = b;
29            b = next;      }
30        System.out.println();   }
31    public double power(double base, int exponent) {
32        return Math.pow(base, exponent);  }}
33 public class MathUtility {
34    public static void main(String[] args) {
35        Scanner scanner = new Scanner(System.in);
36        BasicMath basicMath = new BasicMath();
37        AdvancedMath advancedMath = new AdvancedMath();
38        while (true) {
39            System.out.println("\nMath Operations:");
40            System.out.println("1. Factorial");
41            System.out.println("2. GCD (Greatest Common Divisor)");
42            System.out.println("3. Prime Check");
43            System.out.println("4. Fibonacci Series");
44            System.out.println("5. Exponentiation (Power Calculation)");
45            System.out.println("6. Exit");
46            System.out.print("Choose an option: ");
47            int choice = scanner.nextInt();
48            switch (choice) {
49                case 1:
50                    System.out.print("Enter a number: ");
51                    int num = scanner.nextInt();
52                    System.out.println("Factorial of " + num + " is " + basicMath.factorial(num));
53                    break;
54                case 2:
55                    System.out.print("Enter two numbers: ");
56                    int a = scanner.nextInt();
57                    int b = scanner.nextInt();
58                    System.out.println("GCD of " + a + " and " + b + " is " + basicMath.gcd(a, b));
59                    break;
60                case 3:
61                    System.out.print("Enter a number: ");
62                    int primeCheck = scanner.nextInt();
63                    if (basicMath.isPrime(primeCheck))
64                        System.out.println(primeCheck + " is a Prime Number.");
65                    else
66                        System.out.println(primeCheck + " is NOT a Prime Number.");
67                    break;
68                case 4:
69                    System.out.print("Enter the number of terms: ");
70                    int terms = scanner.nextInt();
71                    advancedMath.fibonacci(terms);
72                    break;
73                case 5:
74                    System.out.print("Enter base and exponent: ");
75                    double base = scanner.nextDouble();
76                    int exponent = scanner.nextInt();
77                    System.out.println(base + "^" + exponent + " = " + advancedMath.power(base, exponent));
78                    break;
79                case 6:
80                    System.out.println("Exiting Math Utility...");
81                    scanner.close();
82                    return;
83                default:
84                    System.out.println("Invalid choice! Please try again.");
85            }}}}

```

Output:

```
Math Operations:  
1. Factorial  
2. GCD (Greatest Common Divisor)  
3. Prime Check  
4. Fibonacci Series  
5. Exponentiation (Power Calculation)  
6. Exit  
Choose an option: 2  
Enter two numbers: 2  
18  
GCD of 2 and 18 is 2  
  
Math Operations:  
1. Factorial  
2. GCD (Greatest Common Divisor)  
3. Prime Check  
4. Fibonacci Series  
5. Exponentiation (Power Calculation)  
6. Exit  
Choose an option: 6  
Exiting Math Utility...
```

13)ABSTRACTION CLASS PROGRAM

13.a)Code:

```

1 import java.util.Scanner;
2
3 abstract class TourPackage {
4     protected String packageName;
5     protected double pricePerPerson;
6
7     public TourPackage(String packageName, double pricePerPerson) {
8         this.packageName = packageName;
9         this.pricePerPerson = pricePerPerson;
10    }
11
12    abstract void showItinerary();
13
14    public double calculateTotalCost(int numPeople, int days) {
15        return numPeople * pricePerPerson * days;
16    }
17 }
18
19 class AdventureTour extends TourPackage {
20     public AdventureTour() {
21         super("Adventure Tour", 100);
22     }
23
24     void showItinerary() {
25         System.out.println("Itinerary: Hiking, Rafting, Camping");
26     }
27 }
28
29 class LuxuryTour extends TourPackage {
30     public LuxuryTour() {
31         super("Luxury Tour", 250);
32     }
33
34     void showItinerary() {
35         System.out.println("Itinerary: 5-Star Hotels, Private Beaches, Fine Dining");
36     }
37 }
38
39 public class TourBookingSystem {
40     public static void main(String[] args) {
41         Scanner scanner = new Scanner(System.in);
42
43         System.out.println("Welcome to the Tour Booking System!");
44         System.out.print("Enter your name: ");
45         String customerName = scanner.nextLine();
46
47         System.out.println("\nSelect a Tour Package:");
48         System.out.println("1. Adventure Tour (100/- per person/day)");
49         System.out.println("2. Luxury Tour (250/- per person/day)");
50         System.out.print("Enter your choice (1 or 2): ");
51         int choice = scanner.nextInt();
52
53         TourPackage selectedPackage;
54         if (choice == 1) {
55             selectedPackage = new AdventureTour();
56         } else if (choice == 2) {
57             selectedPackage = new LuxuryTour();
58         } else {
59             System.out.println("Invalid choice! Exiting...");
60             return;
61         }
62
63         System.out.print("\nEnter number of people: ");
64         int numPeople = scanner.nextInt();
65
66         System.out.print("Enter number of days: ");
67         int numDays = scanner.nextInt();
68
69         System.out.println("\nBooking Details:");
70         System.out.println("Customer Name: " + customerName);
71         System.out.println("Selected Package: " + selectedPackage.packageName);
72         selectedPackage.showItinerary();
73         double totalCost = selectedPackage.calculateTotalCost(numPeople, numDays);
74         System.out.println("Total Cost: " + totalCost+"INR");
75
76         System.out.println("\n Booking Confirmed! Enjoy your trip.");
77         scanner.close();
78     }
79 }
80

```

Output:

```
Welcome to the Tour Booking System!
Enter your name: Raahithya

Select a Tour Package:
1. Adventure Tour (100/- per person/day)
2. Luxury Tour (250/- per person/day)
Enter your choice (1 or 2): 2

Enter number of people: 3
Enter number of days: 4

Booking Details:
Customer Name: Raahithya
Selected Package: Luxury Tour
Itinerary: 5-Star Hotels, Private Beaches, Fine Dining
Total Cost: 3000.0INR

Booking Confirmed! Enjoy your trip.
```

13.b)Code:

```

1 abstract class Device {
2     protected String deviceName;
3     protected boolean isOn;
4
5     public Device(String deviceName) {
6         this.deviceName = deviceName;
7         this.isOn = false;
8     }
9
10    public void turnOn() {
11        isOn = true;
12        System.out.println(deviceName + " is turned ON.");
13    }
14
15    public void turnOff() {
16        isOn = false;
17        System.out.println(deviceName + " is turned OFF.");
18    }
19
20    abstract void performFunction();
21 }
22
23 class SmartLight extends Device {
24     public SmartLight(String deviceName) {
25         super(deviceName);
26     }
27
28     void performFunction() {
29         System.out.println("Adjusting brightness of " + deviceName);
30     }
31 }
32
33 class SmartThermostat extends Device {
34     public SmartThermostat(String deviceName) {
35         super(deviceName);
36     }
37
38     void performFunction() {
39         System.out.println("Adjusting temperature of " + deviceName);
40     }
41 }
42
43 public class HomeAutomation {
44     public static void main(String[] args) {
45         Device light = new SmartLight("Living Room Light");
46         Device thermostat = new SmartThermostat("Bedroom Thermostat");
47
48         light.turnOn();
49         light.performFunction();
50
51         thermostat.turnOn();
52         thermostat.performFunction();
53     }
54 }
```

Output:

Living Room Light is turned ON.
Adjusting brightness of Living Room Light
Bedroom Thermostat is turned ON.
Adjusting temperature of Bedroom Thermostat

13.c)Code:

```

1 abstract class Package {
2     protected String trackingNumber;
3     protected double weight;
4
5     public Package(String trackingNumber, double weight) {
6         this.trackingNumber = trackingNumber;
7         this.weight = weight;
8     }
9
10    abstract double calculateShippingCost();
11
12    public void showDetails() {
13        System.out.println("Tracking Number: " + trackingNumber);
14        System.out.println("Weight: " + weight + "kg");
15    }
16 }
17
18 class StandardPackage extends Package {
19     public StandardPackage(String trackingNumber, double weight) {
20         super(trackingNumber, weight);
21     }
22
23     double calculateShippingCost() {
24         return weight * 10;
25     }
26 }
27
28 class ExpressPackage extends Package {
29     public ExpressPackage(String trackingNumber, double weight) {
30         super(trackingNumber, weight);
31     }
32
33     double calculateShippingCost() {
34         return weight * 20;
35     }
36 }
37
38 public class PackageDelivery {
39     public static void main(String[] args) {
40         Package standard = new StandardPackage("STD12345", 7.5);
41         Package express = new ExpressPackage("EXP67890", 9.2);
42
43         standard.showDetails();
44         System.out.println("Standard Shipping Cost: " + standard.calculateShippingCost());
45
46         express.showDetails();
47         System.out.println("Express Shipping Cost: " + express.calculateShippingCost());
48     }
49 }
```

Output:

Tracking Number: STD12345

Weight: 7.5kg

Standard Shipping Cost: 75.0

Tracking Number: EXP67890

Weight: 9.2kg

Express Shipping Cost: 184.0

13.d)Code:

```

1 import java.util.Scanner;
2
3 abstract class Element {
4     protected String name;
5     protected String symbol;
6     protected int atomicNumber;
7
8     public Element(String name, String symbol, int atomicNumber) {
9         this.name = name;
10        this.symbol = symbol;
11        this.atomicNumber = atomicNumber;
12    }
13
14     abstract void elementProperties();
15
16     public void showDetails() {
17         System.out.println("\nElement: " + name);
18         System.out.println("Symbol: " + symbol);
19         System.out.println("Atomic Number: " + atomicNumber);
20     }
21 }
22
23 class Metal extends Element {
24     public Metal(String name, String symbol, int atomicNumber) {
25         super(name, symbol, atomicNumber);
26     }
27
28     void elementProperties() {
29         System.out.println("Properties: Good conductor of heat and electricity");
30     }
31 }
32
33 class NonMetal extends Element {
34     public NonMetal(String name, String symbol, int atomicNumber) {
35         super(name, symbol, atomicNumber);
36     }
37
38     void elementProperties() {
39         System.out.println("Properties: Poor conductor");
40     }
41 }
42
43 public class ChemicalElements {
44     public static void main(String[] args) {
45         Scanner scanner = new Scanner(System.in);
46
47         Element[] elements = {
48             new Metal("Iron", "Fe", 26),
49             new Metal("Copper", "Cu", 29),
50             new Metal("Aluminum", "Al", 13),
51             new NonMetal("Oxygen", "O", 8),
52             new NonMetal("Carbon", "C", 6),
53             new NonMetal("Sulfur", "S", 16)
54         };
55
56         System.out.print("Enter an element name (Iron, Copper, Aluminum, Oxygen, Carbon, Sulfur): ");
57         String inputName = scanner.nextLine().trim();
58
59         Element foundElement = null;
60         for (Element e : elements) {
61             if (e.name.equalsIgnoreCase(inputName)) {
62                 foundElement = e;
63                 break;
64             }
65         }
66
67         if (foundElement != null) {
68             foundElement.showDetails();
69             foundElement.elementProperties();
70         } else {
71             System.out.println("Element not found! Please enter a valid element name.");
72         }
73
74         scanner.close();
75     }
76 }
```

Output:

```
Enter an element name (Iron, Copper, Aluminum, Oxygen, Carbon, Sulfur): copper
Element: Copper
Symbol: Cu
Atomic Number: 29
Properties: Good conductor of heat and electricity
```

14)ENCAPSULATION PROGRAMS

14.a)Code:

```
1 import java.util.Scanner;
2
3 class Book {
4     private String title;
5     private String author;
6     private double price;
7
8     public Book(String title, String author, double price) {
9         this.title = title;
10        this.author = author;
11        this.price = price;
12    }
13
14    public String getTitle() { return title; }
15    public String getAuthor() { return author; }
16    public double getPrice() { return price; }
17
18    public void setPrice(double price) {
19        if (price > 0) {
20            this.price = price;
21        } else {
22            System.out.println("Invalid price! Price must be positive.");
23        }
24    }
25
26    public void showBookDetails() {
27        System.out.println("Book Title: " + title);
28        System.out.println("Author: " + author);
29        System.out.println("Price: " + price+"/-");
30    }
31 }
32
33 public class LibraryManagement {
34     public static void main(String[] args) {
35         Book book1 = new Book("Harry Potter", "J.K. Rowling", 499.99);
36         book1.showBookDetails();
37
38         book1.setPrice(599.00);
39         System.out.println("\nUpdated Book Price: " + book1.getPrice() +"/-");
40     }
41 }
```

Output:

Book Title: Harry Potter

Author: J.K. Rowling

Price: 499.99/-

Updated Book Price: 599.0/-

14.b)Code:

```

1 import java.util.Scanner;
2
3 class MovieTicket {
4     private String movieName;
5     private int seats;
6     private double ticketPrice;
7
8     public MovieTicket(String movieName, int seats, double ticketPrice) {
9         this.movieName = movieName;
10        setSeats(seats);
11        setTicketPrice(ticketPrice);
12    }
13
14    public String getMovieName() { return movieName; }
15    public int getSeats() { return seats; }
16    public double getTicketPrice() { return ticketPrice; }
17
18    public void setSeats(int seats) {
19        if (seats > 0 && seats <= 10) {
20            this.seats = seats;
21        } else {
22            System.out.println("Invalid seat count! Must be between 1 and 10.");
23        }
24    }
25
26    public void setTicketPrice(double ticketPrice) {
27        if (ticketPrice > 0) {
28            this.ticketPrice = ticketPrice;
29        } else {
30            System.out.println("Invalid ticket price!");
31        }
32    }
33
34    public double calculateTotalCost() {
35        return seats * ticketPrice;
36    }
37
38    public void showBookingDetails() {
39        System.out.println("Movie: " + movieName);
40        System.out.println("Seats Booked: " + seats);
41        System.out.println("Total Cost: " + calculateTotalCost() + "-");
42    }
43 }
44
45 public class MovieBooking {
46     public static void main(String[] args) {
47         Scanner scanner = new Scanner(System.in);
48
49         System.out.print("Enter movie name: ");
50         String name = scanner.nextLine();
51
52         System.out.print("Enter number of seats (Max 10): ");
53         int seats = scanner.nextInt();
54
55         System.out.print("Enter ticket price per seat: ");
56         double price = scanner.nextDouble();
57
58         MovieTicket booking = new MovieTicket(name, seats, price);
59         booking.showBookingDetails();
60
61         scanner.close();
62     }
63 }
```

Output:

```
Enter movie name: Devara
Enter number of seats (Max 10): 4
Enter ticket price per seat: 100
Movie: Devara
Seats Booked: 4
Total Cost: 400.0/-
```

14.c)Code:

```
1 import java.util.Scanner;
2
3 class Course {
4     private String courseName;
5     private String instructor;
6     private double fee;
7
8     public String getCourseName() { return courseName; }
9     public void setCourseName(String courseName) { this.courseName = courseName; }
10
11    public String getInstructor() { return instructor; }
12    public void setInstructor(String instructor) { this.instructor = instructor; }
13
14    public double getFee() { return fee; }
15    public void setFee(double fee) {
16        if (fee > 0) {
17            this.fee = fee;
18        } else {
19            System.out.println("Invalid fee! Must be greater than 0.");
20        }
21    }
22
23    public void showCourseDetails() {
24        System.out.println("\nCourse Details:");
25        System.out.println("Course Name: " + courseName);
26        System.out.println("Instructor: " + instructor);
27        System.out.println("Fee: " + fee +"/-");
28    }
29 }
30
31 public class OnlineCourseEnrollment {
32     public static void main(String[] args) {
33         Scanner scanner = new Scanner(System.in);
34         Course course = new Course();
35
36         System.out.print("Enter Course Name: ");
37         course.setCourseName(scanner.nextLine());
38
39         System.out.print("Enter Instructor Name: ");
40         course.setInstructor(scanner.nextLine());
41
42         System.out.print("Enter Course Fee: ");
43         double fee = scanner.nextDouble();
44         course.setFee(fee);
45
46         course.showCourseDetails();
47
48         scanner.close();
49     }
50 }
```

Output:

```
Enter Course Name: cse
Enter Instructor Name: suthir
Enter Course Fee: 12000

Course Details:
Course Name: cse
Instructor: suthir
Fee: 12000.0/-
```

14.d)Code:

```

1 import java.util.Scanner;
2
3 interface ElectronicDevice {
4     void powerOn();
5     void powerOff();
6     void setVolume(int volume);
7 }
8
9 class Smartphone implements ElectronicDevice {
10    private int volume;
11
12    public void powerOn() {
13        System.out.println("Smartphone is now ON.");
14    }
15
16    public void powerOff() {
17        System.out.println("Smartphone is now OFF.");
18    }
19
20    public void setVolume(int volume) {
21        if (volume >= 0 && volume <= 10) {
22            this.volume = volume;
23            System.out.println("Smartphone Volume set to: " + volume);
24        } else {
25            System.out.println("Invalid Volume! Set between 0 - 10.");
26        }
27    }
28 }
29
30 class Laptop implements ElectronicDevice {
31    private int volume;
32
33    public void powerOn() {
34        System.out.println("Laptop is now ON.");
35    }
36
37    public void powerOff() {
38        System.out.println("Laptop is now OFF.");
39    }
40
41    public void setVolume(int volume) {
42        if (volume >= 0 && volume <= 100) {
43            this.volume = volume;
44            System.out.println("Laptop Volume set to: " + volume);
45        } else {
46            System.out.println("Invalid Volume! Set between 0 - 100.");
47        }
48    }
49 }
50
51 public class Device {
52    public static void main(String[] args) {
53        Scanner scanner = new Scanner(System.in);
54        ElectronicDevice device = null;
55
56        System.out.println("Select Device: 1. Smartphone 2. Laptop");
57        int choice = scanner.nextInt();
58
59        if (choice == 1) {
60            device = new Smartphone();
61        } else if (choice == 2) {
62            device = new Laptop();
63        } else {
64            System.out.println("Invalid Choice!");
65            return;
66        }
67
68        device.powerOn();
69
70        System.out.print("Set Volume: ");
71        int volume = scanner.nextInt();
72        device.setVolume(volume);
73
74        device.powerOff();
75
76        scanner.close();
77    }
78 }

```

Output:

Select Device: 1. Smartphone 2. Laptop

15)PACKAGES

15.a)User Defined Packages

Code:

```

1 package raahi;
2
3 public class Hospital {
4
5     public void registerPatient(String name, int age, String gender) {
6         System.out.println("Patient Registered:");
7         System.out.println("Name: " + name + ", Age: " + age + ", Gender: " + gender);
8     }
9
10    public void assignDoctor(String department) {
11        System.out.println("Doctor assigned to the " + department + " department.");
12    }
13
14    public void dischargePatient(String name) {
15        System.out.println("Patient '" + name + "' has been discharged.");
16    }
17 }
18

```

```

1 package raahi;
2 import java.util.Scanner;
3 public class Main {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         Hospital hospital = new Hospital();
7         System.out.print("Enter patient name: ");
8         String name = scanner.nextLine();
9         System.out.print("Enter age: ");
10        int age = scanner.nextInt();
11        scanner.nextLine();
12        System.out.print("Enter gender: ");
13        String gender = scanner.nextLine();
14        hospital.registerPatient(name, age, gender);
15        System.out.print("Enter department for doctor assignment: ");
16        String department = scanner.nextLine();
17        hospital.assignDoctor(department);
18        System.out.print("Enter patient name to discharge: ");
19        String discharge = scanner.nextLine();
20        hospital.dischargePatient(discharge);
21        scanner.close();
22    }
23 }

```

Output:

```
D:\java programs>javac -d . raahi/Main.java

D:\java programs>java raahi.Main
Enter patient name: raahitya
Enter age: 99
Enter gender: femlae
Patient Registered:
Name: raahitya, Age: 99, Gender: femlae
Enter department for doctor assignment: bp
Doctor assigned to the bp department.
Enter patient name to discharge: raahitya
Patient 'raahitya' has been discharged.
```

15.b)Code:

```
1 package raahi;
2
3 public class Library {
4
5     public void borrowBook(String bookName) {
6         System.out.println("Book '" + bookName + "' has been borrowed.");
7     }
8
9     public void returnBook(String bookName) {
10        System.out.println("Book '" + bookName + "' has been returned.");
11    }
12
13    public void searchBook(String bookName) {
14        System.out.println("Searching for book: " + bookName + "...");
15        System.out.println("Book '" + bookName + "' is available in the library.");
16    }
17 }
```

```
1 package raahi;
2 import java.util.Scanner;
3 public class Main1 {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         Library lib = new Library();
7
8         System.out.print("Enter book name to borrow: ");
9         String borrow = scanner.nextLine();
10        lib.borrowBook(borrow);
11
12        System.out.print("Enter book name to return: ");
13        String ret = scanner.nextLine();
14        lib.returnBook(ret);
15
16        System.out.print("Enter book name to search: ");
17        String search = scanner.nextLine();
18        lib.searchBook(search);
19
20        scanner.close();
21    }
22 }
```

Output:

```
D:\java programs>javac -d . raahi/Main1.java

D:\java programs>java raahi.Main1
Enter book name to borrow: twilight
Book 'twilight' has been borrowed.
Enter book name to return: twited games
Book 'twited games' has been returned.
Enter book name to search: hanutting
Searching for book: hanutting...
Book 'hanutting' is available in the library.
```

15.c)**Code:**

```

1 import java.util.*;
2 import java.io.*;
3
4 public class ReverseNames{
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7         List<String> names = new ArrayList<>();
8
9         System.out.println("Enter names (type 'done' to finish):");
10        while (true) {
11            String name = scanner.nextLine();
12            if (name.equalsIgnoreCase("done")) {
13                break;
14            }
15            names.add(name);
16        }
17
18        Collections.reverse(names);
19        try (BufferedWriter writer = new BufferedWriter(new FileWriter("reversed_names.txt"))) {
20            for (String name : names) {
21                writer.write(name + "\n");
22            }
23            System.out.println("Names saved in reverse order.");
24        } catch (IOException e) {
25            System.err.println("Error saving names: " + e.getMessage());
26        }
27    }
28}
29

```

Output:

```

Enter names (type 'done' to finish):
Raahithya
charitha
srija
purnisha
bhavani
done
Names saved in reverse order.

```

Txt file:

```

bhavani
purnisha
srija
charitha
Raahithya

```

15.d)

Code

```
1 import java.io.*;
2 import java.util.*;
3 import java.text.*;
4
5 public class ItemListSaver {
6     public static void main(String[] args) {
7         List<String> items = Arrays.asList("Apple", "Banana", "Orange", "Grapes");
8
9         try (BufferedWriter writer = new BufferedWriter(new FileWriter("items.txt"))) {
10             for (String item : items) {
11                 writer.write(item);
12                 writer.newLine();
13             }
14         } catch (IOException e) {
15             System.err.println("Error writing to file: " + e.getMessage());
16         }
17
18         String currentDate = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date());
19         System.out.println("Items saved on: " + currentDate);
20
21         try (BufferedReader reader = new BufferedReader(new FileReader("items.txt"))) {
22             String line;
23             while ((line = reader.readLine()) != null) {
24                 System.out.println(line);
25             }
26         } catch (IOException e) {
27             System.err.println("Error reading from file: " + e.getMessage());
28         }
29     }
30 }
31 }
```

Output:

```
Items saved on: 2025-04-04 10:42:53
Apple
Banana
Orange
Grapes
```

Txt file:

```
Apple
Banana
Orange
Grapes
```

16)EXCEPTION HANDLING

16.a)

Code:

```
1 import java.util.Scanner;
2 import java.util.InputMismatchException;
3
4 public class UserInput {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7         try {
8             System.out.print("Enter an integer: ");
9             int number = scanner.nextInt();
10            System.out.println("You entered: " + number);
11        } catch (InputMismatchException e) {
12            System.out.println("Invalid input! Please enter an integer.");
13        } finally {
14            scanner.close();
15        }
16    }
17 }
18 }
```

Output:

```
Enter an integer: Raahi
Invalid input! Please enter an integer.
```

16.b)

Code:

```
1 import java.io.File;
2 import java.io.FileNotFoundException;
3 import java.util.Scanner;
4
5 public class FileNotFoundException{
6     public static void main(String[] args) {
7         File file = new File("filereading.txt");
8
9         try {
10             Scanner scanner = new Scanner(file);
11             while (scanner.hasNextLine()) {
12                 System.out.println(scanner.nextLine());
13             }
14             scanner.close();
15         } catch (FileNotFoundException e) {
16             System.out.println("File not found! Please check the file path.");
17         }
18     }
19 }
20
```

Output:

```
File not found! Please check the file path.
```

16.c)

Code:

```
1 import java.util.Scanner;
2
3 class InvalidLoginException extends Exception {
4     public InvalidLoginException(String message) {
5         super(message);
6     }
7 }
8
9 public class LoginSystem {
10    public static void main(String[] args) {
11        String correctUsername = "admin";
12        String correctPassword = "1234";
13
14        Scanner scanner = new Scanner(System.in);
15
16        try {
17            System.out.print("Enter username: ");
18            String username = scanner.nextLine();
19
20            System.out.print("Enter password: ");
21            String password = scanner.nextLine();
22
23            if (!username.equals(correctUsername) || !password.equals(correctPassword)) {
24                throw new InvalidLoginException("Incorrect username or password!");
25            }
26
27            System.out.println("Login successful!");
28
29        } catch (InvalidLoginException e) {
30            System.out.println("Login failed: " + e.getMessage());
31        } finally {
32            scanner.close();
33        }
34    }
35 }
```

Output:

```
Enter username: Rahi
Enter password: 1234
Login failed: Incorrect username or password!
PS D:\java programs\Exception Handling> javac LoginSystem.java
PS D:\java programs\Exception Handling> java LoginSystem
Enter username: admin
Enter password: 1234
Login successful!
```

16.d)

Code:

```
1 import java.util.Scanner;
2
3 public class DivideByZero{
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         try {
8             System.out.print("Enter numerator: ");
9             int num = scanner.nextInt();
10
11            System.out.print("Enter denominator: ");
12            int den = scanner.nextInt();
13
14            int result = num / den;
15            System.out.println("Result: " + result);
16
17        } catch (ArithmaticException e) {
18            System.out.println("Error: Cannot divide by zero!");
19        } finally {
20            scanner.close();
21        }
22    }
23 }
```

Output:

```
Enter numerator: 3
Enter denominator: 5
Result: 0
PS D:\java programs\Exception Handling> javac DivideByZero.java
PS D:\java programs\Exception Handling> java DivideByZero
Enter numerator: 5
Enter denominator: 0
Error: Cannot divide by zero!
```

17)FILE HANDLING

17.a)

Code:

```
1 import java.io.File;
2
3 public class FileInfo {
4     public static void main(String[] args) {
5         File file = new File("fileInformations.txt");
6         if (file.exists()) {
7             System.out.println("File name: " + file.getName());
8             System.out.println("Absolute path: " + file.getAbsolutePath());
9             System.out.println("File size in bytes: " + file.length());
10            System.out.println("Is directory: " + file.isDirectory());
11        } else {
12            System.out.println("The file does not exist.");
13        }
14    }
15 }
```

Output:

```
File name: fileInformations.txt
Absolute path: D:\java programs\file handling\fileInformations.txt
File size in bytes: 0
Is directory: false
```

17.b)

Code:

```
1 import java.io.FileWriter;
2 import java.io.IOException;
3
4 public class WriteToFile{
5     public static void main(String[] args) {
6         try {
7             FileWriter writer = new FileWriter("filewriting.txt");
8             writer.write("This code is an example of file writing in file handling");
9             writer.close();
10            System.out.println("Successfully wrote info to the file.");
11        } catch (IOException e) {
12            System.out.println("An error found.");
13            e.printStackTrace();
14        }
15    }
16 }
```

Output:

```
Successfully wrote info to the file.
```

Txt file:

```
This code is an example of file| writing in file handling
```

16.c)

Code:

```
1 import java.io.BufferedReader;
2 import java.io.FileReader;
3 import java.io.IOException;
4
5 public class ReadFromFile{
6     public static void main(String[] args) {
7         try {
8             BufferedReader reader = new BufferedReader(new FileReader("filereading.txt"));
9             String line;
10            while ((line = reader.readLine()) != null) {
11                System.out.println(line);
12            }
13            reader.close();
14        } catch (IOException e) {
15            System.out.println("An error occurred.");
16            e.printStackTrace();
17        }
18    }
19 }
```

Output:

```
Hi i'm a student from amrita Vishwa vidyapeetam.
This is my java file handling program
```

16.d)

Code:

```
1 import java.io.*;
2
3 public class FileAppend {
4     public static void main(String[] args) {
5         try {
6             FileWriter fw = new FileWriter("filereading.txt", true);
7             fw.write("\nJava programming is good.");
8             fw.close();
9             System.out.println("Data appended to file successfully.");
10        } catch (IOException e) {
11            System.out.println("An error occurred.");
12            e.printStackTrace();
13        }
14    }
15 }
16 }
```

Output:

```
Data appended to file successfully.
```

Txt file:

```
Hi i'm a student from amrita Vishwa Vidyapeetam.
This is my java file handling program|
Java programming is good.
```