

java lab exam

1.

1.1) Program to remove all repeated elements from an array

without user input

```
import java.util.ArrayList;
import java.util.HashSet;

public class RemoveDuplicatesFromArray {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 4, 3, 2, 1, 5, 6, 7, 5, 4};
        ArrayList<Integer> uniqueElements = new ArrayList<>();
        HashSet<Integer> set = new HashSet<>();

        for (int num : arr) {
            if (!set.contains(num)) {
                uniqueElements.add(num);
                set.add(num);
            }
        }

        // Print the array without duplicate elements
        System.out.println("Array without duplicate elements: " + uniqueElements);
    }
}
```

with user input

```
import java.util.ArrayList;
import java.util.HashSet;
import java.util.Scanner;

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

        System.out.print("Enter the number of elements in the array: ");
        int n = scanner.nextInt();

        int[] arr = new int[n];

        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        ArrayList<Integer> uniqueElements = new ArrayList<>();
        HashSet<Integer> set = new HashSet<>();

        for (int num : arr) {
            if (!set.contains(num)) {
                uniqueElements.add(num);
                set.add(num);
            }
        }

        // Print the array without duplicate elements
    }
}
```

```

        System.out.println("Array without duplicate elements: " + uniqueElements);
    }
}

```

1.2) Write a Java program to find the common elements between two arrays of integers.

without user input

```

import java.util.ArrayList;
import java.util.HashSet;

public class CommonElementsArrays {
    public static void main(String[] args) {
        int[] arr1 = {1, 2, 3, 4, 5};
        int[] arr2 = {4, 5, 6, 7, 8};
        ArrayList<Integer> commonElements = new ArrayList<>();
        HashSet<Integer> set = new HashSet<>();

        for (int num : arr1) {
            set.add(num);
        }

        for (int num : arr2) {
            if (set.contains(num)) {
                commonElements.add(num);
            }
        }

        // Print the common elements between the two arrays
        System.out.println("Common elements: " + commonElements);
    }
}

```

with user input

```

import java.util.ArrayList;
import java.util.HashSet;
import java.util.Scanner;

public class CommonElementsArrays {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements in the first array: ");
        int n1 = scanner.nextInt();

        int[] arr1 = new int[n1];

        System.out.println("Enter the elements of the first array:");
        for (int i = 0; i < n1; i++) {
            arr1[i] = scanner.nextInt();
        }
    }
}

```

```

System.out.print("Enter the number of elements in the second array: ");
int n2 = scanner.nextInt();

int[] arr2 = new int[n2];

System.out.println("Enter the elements of the second array:");
for (int i = 0; i < n2; i++) {
    arr2[i] = scanner.nextInt();
}

ArrayList<Integer> commonElements = new ArrayList<>();
HashSet<Integer> set = new HashSet<>();

for (int num : arr1) {
    set.add(num);
}

for (int num : arr2) {
    if (set.contains(num)) {
        commonElements.add(num);
    }
}

// Print the common elements between the two arrays
System.out.println("Common elements: " + commonElements);
}
}

```

2.

2.1) Java Program to Count Number of Duplicate Words in String

```

import java.util.Arrays;
import java.util.Scanner;

public class CountDuplicateWords {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String inputString = scanner.nextLine();

        // Remove leading and trailing whitespace and convert to lowercase
        inputString = inputString.trim().toLowerCase();

        // Split the string into words
        String[] words = inputString.split("\\\\s+");

        int duplicateCount = 0;

        // Compare each word with the remaining words in the array
        for (int i = 0; i < words.length - 1; i++) {
            if (words[i].equals("")) {
                continue;
            }

            for (int j = i + 1; j < words.length; j++) {
                if (words[i].equals(words[j])) {
                    duplicateCount++;

                    // Mark the duplicate word as empty to avoid counting it again
                    words[j] = "";
                }
            }
        }

        // Print the result
    }
}

```

```

        System.out.println("Number of duplicate words: " + duplicateCount);
    }
}

```

2.2) How to Check if the String Contains 'e' in umbrella

```

    public class StringContainsExample {
    public static void main(String[] args) {
    String word = "umbrella";
    char targetChar = 'e';
    boolean containsChar = word.contains(String.valueOf(targetChar));

    if (containsChar) {
        System.out.println("The string contains the character '" + targetChar + "'.");
    } else {
        System.out.println("The string does not contain the character '" + targetChar + "'.");
    }
    }
}

```

3.

3.1)Java Program to Reverse a String.

```

// Java program to ReverseString using StringBuilder
import java.lang.*;
import java.io.*;
import java.util.*;

// Class of ReverseString
class Main {
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a string");

        String input = sc.nextLine();

        StringBuilder input1 = new StringBuilder();

        // append a string into StringBuilder input1
        input1.append(input);

        // reverse StringBuilder input1
        input1.reverse();

        // print reversed String
        System.out.println(input1);
    }
}

```

without using string rev

```

import java.util.Scanner;

public class ReverseString {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String inputString = scanner.nextLine();

        String reversedString = reverseString(inputString);

        System.out.println("Reversed string: " + reversedString);
    }

    private static String reverseString(String str) {
        char[] charArray = str.toCharArray();
        int left = 0;
        int right = charArray.length - 1;

        while (left < right) {
            // Swap characters at left and right indices
            char temp = charArray[left];
            charArray[left] = charArray[right];
            charArray[right] = temp;

            // Move the left and right indices towards the center
            left++;
            right--;
        }

        return new String(charArray);
    }
}

```

3.2) Write a Java program to check that String is palindrome or not.

```

import java.util.Scanner;

public class PalindromeCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String inputString = scanner.nextLine();

        boolean isPalindrome = checkPalindrome(inputString);

        if (isPalindrome) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
        }
    }

    private static boolean checkPalindrome(String str) {
        int left = 0;
        int right = str.length() - 1;

        while (left < right) {
            // Compare characters at left and right indices
            if (str.charAt(left) != str.charAt(right)) {
                return false; // Characters do not match, not a palindrome
            }
        }
    }
}

```

```

        // Move the left and right indices towards the center
        left++;
        right--;
    }

    return true; // All characters match, palindrome
}
}

```

4. A Company manufactures Vehicles, which could be a Helicopter, a Car, or a Train depending on the customer's demand. Each Vehicle instance has a method called move, which prints on the console the nature of movement of the vehicle. For example, the Helicopter Flies in Air, the Car Drives on Road and the Train Runs on Track. Write a program that accepts input from the user on the kind of vehicle the user wants to order, and the system should print out nature of movement. Implement all Java coding best practices to implement this program.

```

import java.util.Scanner;

interface Vehicle {
    void move();
}

class Helicopter implements Vehicle {
    @Override
    public void move() {
        System.out.println("The Helicopter Flies in Air");
    }
}

class Car implements Vehicle {
    @Override
    public void move() {
        System.out.println("The Car Drives on Road");
    }
}

class Train implements Vehicle {
    @Override
    public void move() {
        System.out.println("The Train Runs on Track");
    }
}

public class VehicleOrderSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the type of vehicle you want to order (Helicopter/Car/Train): ");
        String vehicleType = scanner.nextLine().trim();

        Vehicle vehicle = createVehicle(vehicleType);
        if (vehicle != null) {
            vehicle.move();
        } else {
            System.out.println("Invalid vehicle type!");
        }

        scanner.close();
    }

    private static Vehicle createVehicle(String vehicleType) {
        switch (vehicleType.toLowerCase()) {
            case "helicopter":
                return new Helicopter();
        }
    }
}

```

```

        case "car":
            return new Car();
        case "train":
            return new Train();
        default:
            return null;
    }
}
}

```

5. We have to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The constructor of student A takes the marks in three subjects as its parameters and the marks in four subjects as its parameters for student B. Create an object for each of the two classes and print the percentage of marks for both the students.

Without user Input:

```

abstract class Marks {
    abstract double getPercentage();
}

class A extends Marks {
    private int subject1Marks;
    private int subject2Marks;
    private int subject3Marks;
    public A(int subject1Marks, int subject2Marks, int subject3Marks) {
        this.subject1Marks = subject1Marks;
        this.subject2Marks = subject2Marks;
        this.subject3Marks = subject3Marks;
    }

    @Override
    double getPercentage() {
        double totalMarks = subject1Marks + subject2Marks + subject3Marks;
        return (totalMarks / 300) * 100;
    }
}

class B extends Marks {
    private int subject1Marks;
    private int subject2Marks;
    private int subject3Marks;
    private int subject4Marks;
    public B(int subject1Marks, int subject2Marks, int subject3Marks, int subject4Marks) {
        this.subject1Marks = subject1Marks;
        this.subject2Marks = subject2Marks;
        this.subject3Marks = subject3Marks;
        this.subject4Marks = subject4Marks;
    }

    @Override
    double getPercentage() {
        double totalMarks = subject1Marks + subject2Marks + subject3Marks + subject4Marks;
        return (totalMarks / 400) * 100;
    }
}

public class MarksPercentage {
    public static void main(String[] args) {
        A studentA = new A(80, 90, 95);
        B studentB = new B(85, 92, 88, 90);
    }
}

```

```

System.out.println("Percentage of marks obtained by Student A: " + studentA.getPercentage());
    System.out.println("Percentage of marks obtained by Student B: " + studentB.getPercentage());
}
}

```

```

//With user input
import java.util.Scanner;

abstract class Marks {
    abstract double getPercentage();
}

class A extends Marks {
    private int subject1Marks;
    private int subject2Marks;
    private int subject3Marks;

    public A(int subject1Marks, int subject2Marks, int subject3Marks) {
        this.subject1Marks = subject1Marks;
        this.subject2Marks = subject2Marks;
        this.subject3Marks = subject3Marks;
    }

    @Override
    double getPercentage() {
        double totalMarks = subject1Marks + subject2Marks + subject3Marks;
        return (totalMarks / 300) * 100;
    }
}

class B extends Marks {
    private int subject1Marks;
    private int subject2Marks;
    private int subject3Marks;
    private int subject4Marks;

    public B(int subject1Marks, int subject2Marks, int subject3Marks, int subject4Marks) {
        this.subject1Marks = subject1Marks;
        this.subject2Marks = subject2Marks;
        this.subject3Marks = subject3Marks;
        this.subject4Marks = subject4Marks;
    }

    @Override
    double getPercentage() {
        double totalMarks = subject1Marks + subject2Marks + subject3Marks + subject4Marks;
        return (totalMarks / 400) * 100;
    }
}

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

        System.out.println("Enter marks obtained by Student A (out of 100) in three subjects:");
        int subject1MarksA = scanner.nextInt();
        int subject2MarksA = scanner.nextInt();
        int subject3MarksA = scanner.nextInt();

        System.out.println("Enter marks obtained by Student B (out of 100) in four subjects:");
        int subject1MarksB = scanner.nextInt();
        int subject2MarksB = scanner.nextInt();
        int subject3MarksB = scanner.nextInt();
        int subject4MarksB = scanner.nextInt();

        A studentA = new A(subject1MarksA, subject2MarksA, subject3MarksA);
        B studentB = new B(subject1MarksB, subject2MarksB, subject3MarksB, subject4MarksB);

        System.out.println("Percentage of marks obtained by Student A: " + studentA.getPercentage());
        System.out.println("Percentage of marks obtained by Student B: " + studentB.getPercentage());
    }
}

```



```

        scanner.close();
    }
}

```

6. Write the following code in your editor below:

A class named Arithmetic with a method named add that takes 2 integers as parameters and returns an integer denoting their sum.

A class named Adder that inherits from a superclass named Arithmetic. The main method in the Tester class should print the following: SAMPLE O/P:My superclass is: Arithmetic
42 13 20

```

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

class Adder extends Arithmetic {
    // This class inherits from the Arithmetic superclass
}

public class Tester {
    public static void main(String[] args) {
        Adder adder = new Adder();
        System.out.println("My superclass is: " + adder.getClass().getSuperclass().getName());

        int a = 42;
        int b = 13;
        int sum = adder.add(a, b);
        System.out.println(a + " + " + b + " = " + sum);
    }
}

```

7. You are required to compute the power of a number by implementing a calculator. Create a class My Calculator which consists of a single method long power (int, int). This method takes two integers n and p, as parameters and finds (n)^p. If either or is negative, then the method must throw an exception which says "n or p should not be negative". Also, if both and are zero, then the method must throw an exception which says "n or p should not be negative".

```

import java.util.Scanner;

class MyCalculator {
    public long power(int n, int p) throws Exception {
        if (n < 0 || p < 0) {
            throw new Exception("n or p should not be negative");
        } else if (n == 0 && p == 0) {
            throw new Exception("n and p should not be zero");
        } else {
            return (long) Math.pow(n, p);
        }
    }
}

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

        System.out.print("Enter the value of n: ");
        int n = scanner.nextInt();
    }
}

```

```

        System.out.print("Enter the value of p: ");
        int p = scanner.nextInt();

        try {
            long result = calculator.power(n, p);
            System.out.println("Result: " + result);
        } catch (Exception e) {
            System.out.println("Exception: " + e.getMessage());
        }

        scanner.close();
    }
}

```

8. You are given a phone book that consists of people's names and their phone number. After that you will be given some person's name as query. For each query, print the phone number of that person. Use HashMap to implement it. The first line will have an integer denoting the number of entries in the phone book. Each entry consists of two lines: a name and the corresponding phone number.

After these, there will be some queries. Each query will contain a person's name. Read the queries until end-of-file. Constraints:

A person's name consists of only lower-case English letters and it may be in the format 'first-name last-name' or in the format 'first-name'. Each phone number has exactly 8 digits without any leading zeros. For each case, print "Not found" if the person has no entry in the phone book. Otherwise, print the person's name and phone number.

```

import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

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

        // Create a HashMap to store the phone book entries
        Map<String, String> phoneBook = new HashMap<>();

        System.out.print("Enter the number of contacts you want to create in the phone book: ");
        int numEntries = Integer.parseInt(scanner.nextLine());

        // Read and store the phone book entries
        for (int i = 0; i < numEntries; i++) {
            System.out.print("Enter Name: ");
            String name = scanner.nextLine().trim().toLowerCase();

            System.out.print("Enter Number: ");
            String phoneNumber = scanner.nextLine().trim();

            if (!phoneNumber.matches("\\d{8}")) {
                System.out.println("Invalid phone number format. Phone number should have 8 digits without leading zeros.");
                i--; // Decrement i to re-enter the current entry
                continue;
            }

            phoneBook.put(name, phoneNumber);
        }

        System.out.println("\nEnter names to check if they exist in the phone book:");

        // Read the queries until end-of-file
        while (scanner.hasNextLine()) {
            String query = scanner.nextLine().trim().toLowerCase();

            // Check if the query name exists in the phone book
            if (phoneBook.containsKey(query)) {
                String phoneNumber = phoneBook.get(query);
                System.out.println(query + " " + phoneNumber);
            } else {

```

```

        System.out.println("Not found");
    }
}

scanner.close();
}
}

```

9. GUI

```

//Usually you will require both swing and awt packages
// even if you are working with just swings.
import javax.swing.*;
import java.awt.*;

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

        // Creating the Frame
        JFrame frame = new JFrame("Chat Frame");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 400);

        // Creating the MenuBar and adding components
        JMenuBar mb = new JMenuBar();
        JMenu m1 = new JMenu("FILE");
        JMenu m2 = new JMenu("Help");
        mb.add(m1);
        mb.add(m2);
        JMenuItem m11 = new JMenuItem("Open");
        JMenuItem m22 = new JMenuItem("Save as");
        m1.add(m11);
        m1.add(m22);

        // Creating the panel at bottom and adding components
        JPanel panel = new JPanel(); // the panel is not visible in output
        JLabel label = new JLabel("Enter Text");
        JTextField tf = new JTextField(10); // accepts upto 10 characters
        JButton send = new JButton("Send");
        JButton reset = new JButton("Reset");
        panel.add(label); // Components Added using Flow Layout
        panel.add(tf);
        panel.add(send);
        panel.add(reset);

        // Text Area at the Center
        JTextArea ta = new JTextArea();

        // Adding Components to the frame.
        frame.getContentPane().add(BorderLayout.SOUTH, panel);
        frame.getContentPane().add(BorderLayout.NORTH, mb);
        frame.getContentPane().add(BorderLayout.CENTER, ta);
        frame.setVisible(true);
    }
}

```

10.

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class java
{
    public static void main(String args[])
    {
        Abc obj=new Abc();
    }
}

```

```

    }
}

class Abc implements ActionListener
{
    JLabel l1;
    JTextField t1;
    JLabel l2;
    JTextField t2;
    JButton b1,b2,b3;
    JLabel l3;
    JTextField t3;

    public Abc()
    {
        JFrame j = new JFrame("Number Addition");
        l1=new JLabel("First Number:");
        t1=new JTextField(20);

        l2=new JLabel("Second Number:");
        t2=new JTextField(20);

        l3 = new JLabel("Result");
        t3 = new JTextField(20);

        b1=new JButton("Add");
        b2 = new JButton("Clear");
        b3 = new JButton("Exit");
        // l3=new JLabel("Result:");

        j.add(l1);
        j.add(t1);
        j.add(l2);
        j.add(t2);
        j.add(l3);
        j.add(t3);
        j.add(b1);
        j.add(b2);
        j.add(b3);
        l1.setBounds(100,50,180,50);
        t1.setBounds(250,50,180,50);
        l2.setBounds(100,150,180,50);
        t2.setBounds(250,150,180,50);
        l3.setBounds(100,250,180,50);
        t3.setBounds(250,250,180,50);
        b1.setBounds(150,350,180,50);
        b2.setBounds(300,350,180,50);
        b3.setBounds(400,450,180,50);
        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        j.setLayout(null);
        j.setVisible(true);
        j.setSize(600,600);

        //setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public void actionPerformed(ActionEvent ae)
    {
        if(ae.getSource()== b1){
            int num1=Integer.parseInt(t1.getText());
            int num2=Integer.parseInt(t2.getText());

            int value=num1+num2;
            t3.setText(" "+value);
        }

        if(ae.getSource()== b2){
            t1.setText("");
            t2.setText("");
            t3.setText("");
        }
    }
}

```

```

    if(ae.getSource() == b3){
        System.exit(0);
    }
}
}

```

11.

11.1) Write a Java program that takes a number as input and prints its multiplication table up to 10. Test Data:

Input a number: 8

Expected Output :

8 x 1 = 8

8 x 2 = 16

8 x 3 = 24

...

8 x 10 = 80

```

import java.util.Scanner;

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

        System.out.print("Input a number: ");
        int number = scanner.nextInt();

        for (int i = 1; i <= 10; i++) {
            int result = number * i;
            System.out.println(number + " x " + i + " = " + result);
        }

        scanner.close();
    }
}

```

11.2) Write a java program to check that given number is prime or not.

```

import java.util.Scanner;

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

        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        boolean isPrime = checkPrime(number);

        if (isPrime) {
            System.out.println(number + " is a prime number.");
        } else {
            System.out.println(number + " is not a prime number.");
        }

        scanner.close();
    }

    public static boolean checkPrime(int number) {
        if (number <= 1) {
            return false;
        }
    }
}

```

```

        // Check for divisibility from 2 to square root of the number
        for (int i = 2; i <= Math.sqrt(number); i++) {
            if (number % i == 0) {
                return false; // Number is divisible by i, not prime
            }
        }

        return true; // Number is prime
    }
}

```

12. Write a Java program to display the pattern like a diamond.

Input number of rows (half of the diamond) :7 Expected Output :

```

import java.util.Scanner;

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

        System.out.print("Input number of rows (half of the diamond): ");
        int rows = scanner.nextInt();

        // Upper half of the diamond
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print("**");
            }
            System.out.println();
        }

        // Lower half of the diamond
        for (int i = rows - 1; i >= 1; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print("**");
            }
            System.out.println();
        }

        scanner.close();
    }
}

```

13.

13.1) Program to find the transpose of a given matrix .

```

import java.util.Scanner;

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

        System.out.print("Enter the number of rows: ");
        int rows = scanner.nextInt();

        System.out.print("Enter the number of columns: ");
        int columns = scanner.nextInt();

        int[][] matrix = new int[rows][columns];

        System.out.println("Enter the elements of the matrix:");

        // Read the matrix elements
        for (int i = 0; i < rows; i++) {

```

```

        for (int j = 0; j < columns; j++) {
            matrix[i][j] = scanner.nextInt();
        }
    }

    int[][] transpose = findTranspose(matrix);

    System.out.println("The transpose of the matrix is:");

    // Print the transpose matrix
    for (int i = 0; i < columns; i++) {
        for (int j = 0; j < rows; j++) {
            System.out.print(transpose[i][j] + " ");
        }
        System.out.println();
    }

    scanner.close();
}

public static int[][] findTranspose(int[][] matrix) {
    int rows = matrix.length;
    int columns = matrix[0].length;

    int[][] transpose = new int[columns][rows];

    // Compute the transpose of the matrix
    for (int i = 0; i < columns; i++) {
        for (int j = 0; j < rows; j++) {
            transpose[i][j] = matrix[j][i];
        }
    }

    return transpose;
}
}

```

13.2) Write Java Program to find the number of the words in the given text file.

```

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

public class WordCount {
    public static void main(String[] args) {
        try {
            // Provide the path to your text file
            File file = new File("path/to/your/text/file.txt");
            Scanner scanner = new Scanner(file);

            int wordCount = 0;

            while (scanner.hasNext()) {
                String word = scanner.next();
                wordCount++;
            }

            System.out.println("Number of words in the text file: " + wordCount);

            scanner.close();
        } catch (FileNotFoundException e) {
            System.out.println("File not found.");
        }
    }
}

```

14.

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class java
{
    public static void main(String args[])
    {
        Abc obj=new Abc();
    }
}

class Abc implements ActionListener
{
    JLabel l1;
    JTextField t1;
    JLabel l2;
    JTextField t2;
    JButton b1,b2,b3;
    JLabel l3;
    JTextField t3;

    public Abc()
    {
        JFrame j = new JFrame("Number Addition");
        l1=new JLabel("First Number:");
        t1=new JTextField(20);

        l2=new JLabel("Second Number:");
        t2=new JTextField(20);

        l3 = new JLabel("Result");
        t3 = new JTextField(20);

        b1=new JButton("Add");
        b2 = new JButton("Clear");
        b3 = new JButton("Exit");
        // l3=new JLabel("Result:");

        j.add(l1);
        j.add(t1);
        j.add(l2);
        j.add(t2);
        j.add(l3);
        j.add(t3);
        j.add(b1);
        j.add(b2);
        j.add(b3);
        l1.setBounds(100,50,180,50);
        t1.setBounds(250,50,180,50);
        l2.setBounds(100,150,180,50);
        t2.setBounds(250,150,180,50);
        l3.setBounds(100,250,180,50);
        t3.setBounds(250,250,180,50);
        b1.setBounds(150,350,180,50);
        b2.setBounds(300,350,180,50);
        b3.setBounds(400,450,180,50);
        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        j.setLayout(null);
        j.setVisible(true);
        j.setSize(600,600);

        //setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public void actionPerformed(ActionEvent ae)
    {
        if(ae.getSource()== b1){
            int num1=Integer.parseInt(t1.getText());
            int num2=Integer.parseInt(t2.getText());

```



```

        int value=num1+num2;
        t3.setText(" "+value);
    }

    if(ae.getSource()== b2){
        t1.setText("");
        t2.setText("");
        t3.setText("");
    }

    if(ae.getSource() == b3){
        System.exit(0);
    }
}

}

```

14. Write a Java Program to iterate ArrayList using for-loop, iterator, and advance for-loop. Insert 3 Array List.Input 20 30 40Output:

```

import java.util.ArrayList;
import java.util.Iterator;

public class ArrayListIteration {
    public static void main(String[] args) {
        // Create an ArrayList
        ArrayList<Integer> numbers = new ArrayList<>();

        // Insert elements into the ArrayList
        numbers.add(20);
        numbers.add(30);
        numbers.add(40);

        System.out.println("Iterator Loop:");
        // Iterate using iterator
        Iterator<Integer> iterator = numbers.iterator();
        while (iterator.hasNext()) {
            System.out.println(iterator.next());
        }

        System.out.println("Advanced For Loop:");
        // Iterate using advanced for-loop
        for (int number : numbers) {
            System.out.println(number);
        }

        System.out.println("For Loop:");
        // Iterate using for-loop
        for (int i = 0; i < numbers.size(); i++) {
            System.out.println(numbers.get(i));
        }
    }
}

```

Output:

iterator Loop:

20

30

40

Advanced For Loop:

20

30

40

For Loop:

20

30

40

16. Write a Java Program to count the number of words in a string using HashMap.Output:

Input :Enter String: "This this is is done by Saket Saket";

{Saket=2, by=1, this=1, This=1, is=2, done=1}

```
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in)(http://system.in/));
        System.out.print("Enter String: ");
        String input = scanner.nextLine();
        scanner.close();
        Map<String, Integer> wordCountMap = countWords(input);
        System.out.println(wordCountMap);
    }

    private static Map<String, Integer> countWords(String input) {
        Map<String, Integer> wordCountMap = new HashMap<>();
        String[] words = input.split(" ");

        for (String word : words) {
            wordCountMap.put(word, wordCountMap.getOrDefault(word, 0) + 1);
        }

        return wordCountMap;
    }
}
//Input:This this is is done by Saket Saket
```

17. Write a program to read 10 string from console and then print the sorted strings on console (Use String Class).2) combine two string 3)reverse first string nd dispaly it .

```
import java.util.Arrays;
import java.util.Scanner;

public class StringOperations {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String[] strings = new String[10];

        // Read 10 strings from console
        System.out.println("Enter 10 strings:");
        for (int i = 0; i < 10; i++) {
            strings[i] = scanner.nextLine();
        }

        // Sort the strings
        Arrays.sort(strings);

        System.out.println("Sorted strings:");
        for (String str : strings) {
            System.out.println(str);
        }
    }
}
```

```

        // Combine two strings
        System.out.println("Combine two strings:");
        String combined = strings[0] + strings[1];
        System.out.println(combined);

        // Reverse the first string
        System.out.println("Reverse first string:");
        String reversed = reverseString(strings[0]);
        System.out.println(reversed);

        scanner.close();
    }

    public static String reverseString(String str) {
        StringBuilder reversed = new StringBuilder(str);
        return reversed.reverse().toString();
    }
}

```

18. Write a program to implement following inheritance. Accept data for 5 persons and display the name of employee having salary greater than 5000.

Class Name: Person

Member variables:

Name, age

Class Name: Employee

Member variables:

Designation, salary

```

import java.util.Scanner;

class Person {
    protected String name;
    protected int age;

    public void acceptData() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter name:");
        name = scanner.nextLine();
        System.out.println("Enter age:");
        age = scanner.nextInt();
    }
}

class Employee extends Person {
    private String designation;
    private double salary;

    public void acceptData() {
        super.acceptData();
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter designation:");
        designation = scanner.nextLine();
        System.out.println("Enter salary:");
        salary = scanner.nextDouble();
    }

    public void displayData() {
        if (salary > 5000) {
            System.out.println("Name: " + name);
        }
    }
}

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

```

```

        Employee[] employees = new Employee[5];

        for (int i = 0; i < 5; i++) {
            System.out.println("Enter details for Employee " + (i + 1));
            employees[i] = new Employee();
            employees[i].acceptData();
        }

        System.out.println("Employees with salary greater than 5000:");
        for (Employee employee : employees) {
            employee.displayData();
        }
    }
}

```

19. Implementing "Multiple Inheritance". Create a two interfaces Account containing methods set() and display() And interface Person containing methods store() and disp(). Derive a class Customer from Person and Account. Accept the name, account number, balance and display all the information related to account along with the interest.

```

import java.util.Scanner;

interface Account {
    void set(String accountNumber, double balance);
}

interface Person {
    void store(String name);
    void disp();
}

class Customer implements Account, Person {
    private String name;
    private String accountNumber;
    private double balance;

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

    public void display() {
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Balance: $" + balance);
    }

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

    public void disp() {
        System.out.println("Name: " + name);
    }

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

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

        System.out.println("Enter name: ");
        String name = scanner.nextLine();
        customer.store(name);
    }
}

```

```

        System.out.println("Enter account number: ");
        String accountNumber = scanner.nextLine();

        System.out.println("Enter balance: ");
        double balance = scanner.nextDouble();

        customer.set(accountNumber, balance);

        System.out.println("\nCustomer Details:");
        customer.disp();
        customer.display();

        System.out.println("\nEnter interest rate: ");
        double interestRate = scanner.nextDouble();
        customer.calculateInterest(interestRate);

        scanner.close();
    }
}

```

20. "Write a program, to implement the following hierarchy. Displays information of each class the rectangle represents the classes. The classes Movie and MusicVideo inherits all the members of the class VideoTape."

without user input:

```

class VideoTape {
    protected String title;
    protected double duration;

    public VideoTape(String title, double duration) {
        this.title = title;
        this.duration = duration;
    }

    public void display() {
        System.out.println("Video Tape Information");
        System.out.println("Title: " + title);
        System.out.println("Duration: " + duration + " minutes");
    }
}

class Movie extends VideoTape {
    private String director;
    private int releaseYear;

    public Movie(String title, double duration, String director, int releaseYear) {
        super(title, duration);
        this.director = director;
        this.releaseYear = releaseYear;
    }

    public void display() {
        super.display();
        System.out.println("Movie Information");
        System.out.println("Director: " + director);
        System.out.println("Release Year: " + releaseYear);
    }
}

class MusicVideo extends VideoTape {
    private String artist;
    private String album;

    public MusicVideo(String title, double duration, String artist, String album) {
        super(title, duration);
        this.artist = artist;
        this.album = album;
    }
}

```

```

        public void display() {
            super.display();
            System.out.println("Music Video Information");
            System.out.println("Artist: " + artist);
            System.out.println("Album: " + album);
        }
    }

    public class ClassHierarchyExample {
        public static void main(String[] args) {
            VideoTape videoTape = new VideoTape("Generic Video", 120.5);
            videoTape.display();

            System.out.println();

            Movie movie = new Movie("The Shawshank Redemption", 142.3, "Frank Darabont", 1994);
            movie.display();

            System.out.println();

            MusicVideo musicVideo = new MusicVideo("Bohemian Rhapsody", 6.2, "Queen", "A Night at the Opera");
            musicVideo.display();
        }
    }
}

```

21. Write a Java program to create a class called "Student" with a name, grade, and courses attributes, and methods to add and remove courses.

```

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

class Student {
    private String name;
    private String grade;
    private List<String> courses;

    public Student(String name, String grade) {
        this.name = name;
        this.grade = grade;
        this.courses = new ArrayList<>();
    }

    public void addCourse(String course) {
        courses.add(course);
        System.out.println("Course added: " + course);
    }

    public void removeCourse(String course) {
        if (courses.contains(course)) {
            courses.remove(course);
            System.out.println("Course removed: " + course);
        } else {
            System.out.println("Course not found: " + course);
        }
    }

    public void displayCourses() {
        System.out.println("Courses for " + name + ":");
        for (String course : courses) {
            System.out.println(course);
        }
    }
}

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

        System.out.println("Enter student name:");
    }
}

```

```

String name = scanner.nextLine();

System.out.println("Enter student grade:");
String grade = scanner.nextLine();

Student student = new Student(name, grade);

while (true) {
    System.out.println("\nMenu:");
    System.out.println("1. Add course");
    System.out.println("2. Remove course");
    System.out.println("3. Display courses");
    System.out.println("4. Quit");
    System.out.println("Enter your choice:");

    int choice = scanner.nextInt();
    scanner.nextLine(); // Consume the newline character

    switch (choice) {
        case 1:
            System.out.println("Enter course name to add:");
            String addCourse = scanner.nextLine();
            student.addCourse(addCourse);
            break;
        case 2:
            System.out.println("Enter course name to remove:");
            String removeCourse = scanner.nextLine();
            student.removeCourse(removeCourse);
            break;
        case 3:
            student.displayCourses();
            break;
        case 4:
            System.out.println("Program terminated.");
            scanner.close();
            return;
        default:
            System.out.println("Invalid choice. Please try again.");
    }
}
}
}

```

22. Write a Java program to create a class known as Person with methods called `getFirstName()` and `getLastName()`. Create a subclass called Employee that adds a new method named `getEmployeeId()` and overrides the `getLastName()` method to include the employee's job title.

Without using user input

```

class Person {
    private String firstName;
    private String lastName;

    public Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }
}

class Employee extends Person {
    private String employeeId;

```

```

    private String jobTitle;

    public Employee(String firstName, String lastName, String employeeId, String jobTitle) {
        super(firstName, lastName);
        this.employeeId = employeeId;
        this.jobTitle = jobTitle;
    }

    public String getEmployeeId() {
        return employeeId;
    }

    @Override
    public String getLastName() {
        return super.getLastName() + " (" + jobTitle + ")";
    }
}

public class PersonEmployeeExample {
    public static void main(String[] args) {
        Person person = new Person("John", "Doe");
        System.out.println("First Name: " + person.getFirstName());
        System.out.println("Last Name: " + person.getLastName());

        System.out.println();

        Employee employee = new Employee("Jane", "Smith", "E12345", "Manager");
        System.out.println("First Name: " + employee.getFirstName());
        System.out.println("Last Name: " + employee.getLastName());
        System.out.println("Employee ID: " + employee.getEmployeeId());
    }
}

```

With user input

```

import java.util.Scanner;

class Person {
    private String firstName;
    private String lastName;

    public Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }
}

class Employee extends Person {
    private String employeeId;
    private String jobTitle;

    public Employee(String firstName, String lastName, String employeeId, String jobTitle) {
        super(firstName, lastName);
        this.employeeId = employeeId;
        this.jobTitle = jobTitle;
    }

    public String getEmployeeId() {
        return employeeId;
    }

    @Override

```



```

    public String getLastName() {
        return super.getLastName() + " (" + jobTitle + ")";
    }
}

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

        System.out.print("Enter person's first name: ");
        String firstName = scanner.nextLine();

        System.out.print("Enter person's last name: ");
        String lastName = scanner.nextLine();

        Person person = new Person(firstName, lastName);
        System.out.println("First Name: " + person.getFirstName());
        System.out.println("Last Name: " + person.getLastName());

        System.out.println();

        System.out.print("Enter employee's first name: ");
        firstName = scanner.nextLine();

        System.out.print("Enter employee's last name: ");
        lastName = scanner.nextLine();

        System.out.print("Enter employee ID: ");
        String employeeId = scanner.nextLine();

        System.out.print("Enter employee's job title: ");
        String jobTitle = scanner.nextLine();

        Employee employee = new Employee(firstName, lastName, employeeId, jobTitle);
        System.out.println("First Name: " + employee.getFirstName());
        System.out.println("Last Name: " + employee.getLastName());
        System.out.println("Employee ID: " + employee.getEmployeeId());

        scanner.close();
    }
}

```

23. Write a Java program to find the length of the longest consecutive elements sequence from an unsorted array of integers.

Sample array: [49, 1, 3, 200, 2, 4, 70, 5]

The longest consecutive elements sequence is [1, 2, 3, 4, 5], therefore the program will return its length 5.

```

import java.util.*;

public class Main {
    public static int longestConsecutive(int[] nums) {
        if (nums == null || nums.length == 0) {
            return 0;
        }

        Arrays.sort(nums); // Sort the array in ascending order
        System.out.println(" " + Arrays.toString(nums));
        int maxLength = 1;
        int currentLength = 1;

        for (int i = 1; i < nums.length; i++) {
            if (nums[i] != nums[i - 1]) {
                // Check if the current element is consecutive to the previous element
                if (nums[i] == nums[i - 1] + 1) {
                    currentLength++;
                } else {
                    maxLength = Math.max(maxLength, currentLength);
                }
            }
        }
    }
}

```

```

        currentLength = 1;
    }
    }else{
        maxLength = Math.max(maxLength, currentLength);
        currentLength = 1;
    }
}

// Check the length of the last sequence
maxLength = Math.max(maxLength, currentLength);

return maxLength;
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter size of array:");
    int n = sc.nextInt();
    int[] nums = new int[n];
    System.out.println("Enter elements: ");
    for(int i=0;i<n;i++){
        nums[i]=sc.nextInt();
    }

    int length = longestConsecutive(nums);

    System.out.println("The length of the longest consecutive sequence is: " + length);
}
}

```

24. Create a class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate user-defined exception "AgeNotWithinRangeException". If name contains numbers or special symbols raise exception "NameNotValidException". Define the two exception classes.

```

import java.util.Scanner;

class AgeNotWithinRangeException extends Exception {
    public AgeNotWithinRangeException(String message) {
        super(message);
    }
}

class NameNotValidException extends Exception {
    public NameNotValidException(String message) {
        super(message);
    }
}

class Student {
    private int rollNo;
    private String name;
    private int age;
    private String course;

    public Student(int rollNo, String name, int age, String course) throws AgeNotWithinRangeException, NameNotValidException {
        if (age < 15 || age > 21) {
            throw new AgeNotWithinRangeException("Age is not within the range of 15 to 21");
        }

        if (!name.matches("[a-zA-Z\\s]+$")) {
            throw new NameNotValidException("Name contains numbers or special symbols");
        }

        this.rollNo = rollNo;
        this.name = name;
        this.age = age;
        this.course = course;
    }
}

```

```

        public void displayInfo() {
            System.out.println("Roll No: " + rollNo);
            System.out.println("Name: " + name);
            System.out.println("Age: " + age);
            System.out.println("Course: " + course);
        }
    }

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

            try {
                System.out.print("Enter Roll No: ");
                int rollNo = scanner.nextInt();

                scanner.nextLine(); // Consume the newline character

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

                System.out.print("Enter Age: ");
                int age = scanner.nextInt();

                scanner.nextLine(); // Consume the newline character

                System.out.print("Enter Course: ");
                String course = scanner.nextLine();

                Student student = new Student(rollNo, name, age, course);
                student.displayInfo();
            } catch (AgeNotWithinRangeException e) {
                System.out.println("Exception: " + e.getMessage());
            } catch (NameNotValidException e) {
                System.out.println("Exception: " + e.getMessage());
            }

            scanner.close();
        }
    }
}

```

25.

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class _25 implements ActionListener {
    public static JButton btn1;
    public static JLabel txt1;
    public static JButton btn2;
    public static JLabel txt2;
    public static JButton btn3;
    public static JLabel txt3;
    public static JButton btn4;
    public static JLabel txt4;
    public static JButton btn5;
    public static JLabel txt5;
    public static JButton btn6;

    public static void main(String[] args) {
        JFrame frame = new JFrame();
        frame.setSize(700, 500);

        JPanel panel = new JPanel();
        panel.setLayout(null);

        btn1 = new JButton("All The Best");
    }
}

```

```

btn1.setBounds(50, 20, 150, 25);
btn1.setForeground(Color.RED);
btn1.addActionListener(new _25());
panel.add(btn1);

txt1 = new JLabel("All The Best");
txt1.setBounds(100, 100, 150, 25);
txt1.setForeground(Color.RED);
panel.add(txt1);

btn2 = new JButton("All The Best");
btn2.setBounds(200, 20, 150, 25);
btn2.setForeground(Color.pink);
btn2.addActionListener(new _25());

txt2 = new JLabel("All The Best");
txt2.setBounds(200, 150, 150, 25);
txt2.setForeground(Color.pink);
panel.add(txt2);
panel.add(btn2);

btn3 = new JButton("All The Best");
btn3.setBounds(350, 20, 150, 25);
btn3.setForeground(Color.blue);
btn3.addActionListener(new _25());

txt3 = new JLabel("All The Best");
txt3.setBounds(300, 200, 150, 25);
txt3.setForeground(Color.blue);
panel.add(txt3);

panel.add(btn3);

btn4 = new JButton("All The Best");
btn4.setBounds(500, 20, 150, 25);
btn4.setForeground(Color.green);
btn4.addActionListener(new _25());

txt4 = new JLabel("All The Best");
txt4.setBounds(400, 250, 150, 25);
txt4.setForeground(Color.green);
panel.add(txt4);

panel.add(btn4);

btn5 = new JButton("All The Best");
btn5.setBounds(275, 50, 150, 25);
btn5.setForeground(Color.cyan);
btn5.addActionListener(new _25());

txt5 = new JLabel("All The Best");
txt5.setBounds(500, 300, 150, 25);
txt5.setForeground(Color.cyan);
panel.add(btn5);
panel.add(txt5);

btn6 = new JButton("Clear");
btn6.setBounds(600, 400, 100, 25);
btn6.setForeground(Color.black);
btn6.addActionListener(new _25());
panel.add(btn6);

txt1.setVisible(false);
txt2.setVisible(false);
txt3.setVisible(false);
txt4.setVisible(false);
txt5.setVisible(false);
frame.add(panel);
frame.setResizable(false);
frame.setVisible(true);

```

```

}

```

```

@Override
public void actionPerformed(ActionEvent e) {
    if(e.getSource() == btn6){
        txt1.setVisible(false);
        txt2.setVisible(false);
        txt3.setVisible(false);
        txt4.setVisible(false);
        txt5.setVisible(false);

    }
    JButton[] btns = { btn1, btn2, btn3, btn4, btn5 };
    JLabel[] txts = { txt1, txt2, txt3, txt4, txt5 };
    for (int i = 0; i < 5; i++) {
        if (e.getSource() == btns[i]) {
            txts[i].setVisible(true);
        }
    }
}
}

```