1.

1.1) Program to remove all repeated elements from an array

import java.util.Scanner;

class RemoveDuplicate{

    public static void main(String[] args) {

        int n;

        Scanner sc=new Scanner(System.in);

        System.out.print("Enter size of array:");

        n=sc.nextInt();

        int [] arr=new int [n];

        System.out.println("Enter the values of array:");

        for(int i=0;i<n;i++){

            arr[i]=sc.nextInt();

        }

        for(int i=0;i<n;i++){

            for(int j=i+1;j<n;j++){

                if(arr[i]==arr[j]){

                    for(int k=j;k<n-1;k++){

                        arr[k]=arr[k+1];

                    }

                    n--;

                    j--;

                }

            }

        }

        for(int i=0;i<n;i++){

            System.out.print(arr[i]+" ");

        }

        }

}

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

public class CommonEle {

    public static void main(String[] args) {

        int arr1[]={1,2,3,4,5};

        int arr2[]={1,5,3,8,9};

        for(int i=0;i<5;i++){

            for(int j=0;j<5;j++){

                if(arr1[i]==arr2[j]){

                    System.out.print(arr1[i]+" ");

                }

            }

        }

    }

}

2.

2. 1) Java Program to Count Number of Duplicate Words in String

import java.util.\*;

public class countDuplicate {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        String input = sc.nextLine();

        String[] words = input.split(" ");

        Map<String, Integer> wordCountMap = new HashMap<>();

        for (String word : words) {

            if (wordCountMap.containsKey(word)) {

                int count = wordCountMap.get(word);

                wordCountMap.put(word, count + 1);

            } else {

                wordCountMap.put(word, 1);

            }

        }

        Iterator<String> it = wordCountMap.keySet().iterator();

        while (it.hasNext()) {

            String key = it.next();

            int count = wordCountMap.get(key);

            if (count > 1) {

                System.out.println(key + " - " + count);

            }

        }

    }

}

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

public class check {

    public static void main(String[] args) {

        String s="umbrela";

        // System.out.println(s.contains("e"));

        boolean ans=false;

        for(int i=0;i<s.length();i++){

            char a=s.charAt(i);

           if(a=='e'){

            ans=true;

           }

        }

        if(ans==true){

            System.out.println("true");

        }

        else{

            System.out.println("false");

        }

    }

}

3.

3.1)Java Program to Reverse a String.

import java.util.\*;

public class reverseString {

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);

        String s=sc.nextLine();

        char temp;

        // System.out.println(s.charAt(j));

        String ans="";

        for(int i=0;i<s.length();i++){

            temp=s.charAt(i);

            ans=temp+ans;

        }

        System.out.println(ans);

    }

}

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

public class checkForPlidromin {

    public static void main(String[] args) {

        String s="aa";

        int i=0;

        boolean ans=true;

        int j=s.length()-1;

        while(i<=j){

            if(s.charAt(i)==s.charAt(j)){

                ans=true;

                i++;

                j--;

            }

            else{

                ans=false;

                break;

            }

        }

        System.out.println(ans);

    }

}

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.\*;

public class comany {

    interface Vehical{

         void move();

    }

    public static class Helicopter implements Vehical{

        @Override

        public void move(){

            System.out.println("The Helicopter Flies in Air");

        }

    }

    public static class Car implements Vehical{

        @Override

        public void move(){

            System.out.println("The Car drives on road");

        }

    }

    public static class Train implements Vehical{

        @Override

        public void move(){

            System.out.println("The Train Runs on Track");

        }

    }

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);

        System.out.println("Welcome to the Vehicle Manufacturing Company");

        System.out.println("What kind of vehicle would you like to order?");

        System.out.println("1. Helicopter");

        System.out.println("2. Car");

        System.out.println("3. Train");

        int choice = sc.nextInt();

        switch(choice){

            case 1:

            Vehical vehical1=new Helicopter();

            vehical1.move();

            break;

            case 2:

            Vehical vehical2=new Car();

            vehical2.move();

            break;

            case 3:

            Vehical vehical3=new Train();

            vehical3.move();

            break;

            default:

            System.out.println("Invalid Optation.");

            break;

        }

        System.out.println("Thnak You !! Have a nice day.");

    }

}

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 eac of the two classes and print the percentage of marks for both the students.

public class marksOfStudent {

    public static abstract class Marks {

        public abstract double getPercentage();

    }

    public static class A extends Marks {

        private double subject1, subject2, subject3;

        public A(double s1, double s2, double s3) {

            subject1 = s1;

            subject2 = s2;

            subject3 = s3;

        }

        @Override

        public double getPercentage() {

            return (subject1 + subject2 + subject3) / 3;

        }

    }

    public static class B extends Marks {

        private double subject1, subject2, subject3, subject4;

        public B(double s1, double s2, double s3, double s4) {

            subject1 = s1;

            subject2 = s2;

            subject3 = s3;

            subject4 = s4;

        }

        @Override

        public double getPercentage() {

            return (subject1 + subject2 + subject3 + subject4) / 4;

        }

    }

    public static void main(String[] args) {

        A studentA = new A(75, 80, 90);

        B studentB = new B(70, 80, 85, 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()+" %");

    }

}

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**

public class sumOfNumber {

    public static class Arithmetic {

        public int add(int a, int b) {

            return a + b;

        }

    }

    public static class Adder extends Arithmetic {

        // empty class body as we are not adding any new functionality

    }

        public static void main(String[] args) {

            Adder adder = new Adder();

            System.out.println("My superclass is: " + adder.getClass().getSuperclass());

            int a = 42, b = 13;

            System.out.println(a + " " + b + " " + adder.add(a, b));

        }

}

**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”.

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 Calculator {

    public static void main(String[] args) {

        MyCalculator calculator = new MyCalculator();

        try {

            long result1 = calculator.power(2, 3);

            System.out.println("Result 1: " + result1);

        } catch (Exception e) {

            System.out.println("Exception: " + e.getMessage());

        }

        try {

            long result2 = calculator.power(-2, 3);

            System.out.println("Result 2: " + result2);

        } catch (Exception e) {

            System.out.println("Exception: " + e.getMessage());

        }

        try {

            long result3 = calculator.power(2, -3);

            System.out.println("Result 3: " + result3);

        } catch (Exception e) {

            System.out.println("Exception: " + e.getMessage());

        }

        try {

            long result4 = calculator.power(0, 0);

            System.out.println("Result 4: " + result4);

        } catch (Exception e) {

            System.out.println("Exception: " + e.getMessage());

        }

    }

}

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);

        System.out.println("Enter number of entry count: ");

        int numEntries = scanner.nextInt();

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

        // Create a HashMap to store the phone book entries

        Map<String, String> phoneBook = new HashMap<>();

        // Read the phone book entries

        for (int i = 0; i < numEntries; i++) {

            System.out.print("Enter name: ");

            String name = scanner.nextLine();

            System.out.print("Enter phone number: ");

            String phoneNumber = scanner.nextLine();

            phoneBook.put(name, phoneNumber);

        }

        // Process the queries

        System.out.print("Enter name for finding its number: ");

        while (scanner.hasNextLine()) {

            String query = scanner.nextLine();

            if (phoneBook.containsKey(query)) {

                String phoneNumber = phoneBook.get(query);

                System.out.println(query + " " + phoneNumber);

            } else {

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

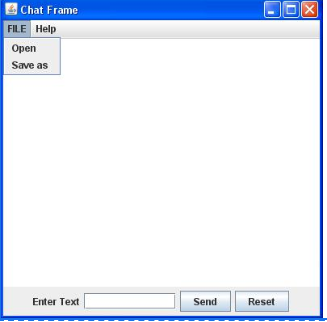
            }

        }

        scanner.close();

    }

}

9. 

import javax.swing.\*;  
  
  
import java.awt.\*;  
  
  
class gui {  
  
  
 public static void main(String args[]) {  
  
  
  
  
 //Create the Frame  
  
  
 JFrame jframe = new JFrame("Chat Screen");  
  
  
 jframe.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
  
 jframe.setSize(400, 400);  
  
  
  
  
// create two menubar button FILE and HELP  
  
  
 JMenuBar menuBar = new JMenuBar();  
  
  
 JMenu fileMenu = new JMenu("FILE");  
  
  
 JMenu helpMenu = new JMenu("Help");  
  
  
 menuBar.add(fileMenu);  
  
  
 menuBar.add(helpMenu);  
  
  
  
  
// create two more option in FILE button  
  
  
 JMenuItem fileMenu1 = new JMenuItem("new file");  
  
  
 JMenuItem fileMenu2 = new JMenuItem("Save as");  
  
  
 fileMenu.add(fileMenu1);  
  
  
 fileMenu.add(fileMenu2);  
  
  
  
  
 // Text Area at the Center  
  
  
 JTextArea textArea = new JTextArea();  
  
  
  
  
 //Create the panel at bottom and add label, textArea and buttons  
  
  
 JPanel panel = new JPanel(); // this panel is not visible in output  
  
  
 JLabel label = new JLabel("Please Enter Text");  
  
  
 JTextField textField = new JTextField(15); // accepts upto 15 characters  
  
  
 JButton btn\_send = new JButton("Send");  
  
  
 JButton btn\_reset = new JButton("Reset");  
  
  
 panel.add(label); // Components Added using Flow Layout  
  
  
 panel.add(textField);  
  
  
 panel.add(btn\_send);  
  
  
 panel.add(btn\_reset);  
  
  
  
  
  
  
  
  
 //Adding Components to the frame.  
  
  
 jframe.getContentPane().add(BorderLayout.*SOUTH*, panel);  
  
  
 jframe.getContentPane().add(BorderLayout.*NORTH*, menuBar);  
  
  
 jframe.getContentPane().add(BorderLayout.*CENTER*, textArea);  
  
  
 jframe.setVisible(true);  
  
  
 }  
  
  
}

10. 

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class NumberAddition {

    private JFrame frame;

    private JTextField firstNumberField;

    private JTextField secondNumberField;

    private JLabel resultLabel;

    public NumberAddition() {

        // Create the frame

        frame = new JFrame("GUI Example");

        frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        frame.setSize(300, 200);

        frame.setLayout(new GridLayout(5,2));

        // Create and add components to the frame

        JLabel firstNumberLabel = new JLabel("First Number:");

        firstNumberField = new JTextField();

        JLabel secondNumberLabel = new JLabel("Second Number:");

        secondNumberField = new JTextField();

        JLabel resultTextLabel = new JLabel("Result:");

        resultLabel = new JLabel();

        frame.add(firstNumberLabel);

        frame.add(firstNumberField);

        frame.add(secondNumberLabel);

        frame.add(secondNumberField);

        frame.add(resultTextLabel);

        frame.add(resultLabel);

        JButton addButton = new JButton("Add");

        addButton.addActionListener(new AddButtonListener());

        JButton clearButton = new JButton("Clear");

        clearButton.addActionListener(new ClearButtonListener());

        JButton exitButton = new JButton("Exit");

        exitButton.addActionListener(new ExitButtonListener());

        frame.add(addButton);

        frame.add(clearButton);

        frame.add(exitButton);

        // Display the frame

        frame.setVisible(true);

    }

    private class AddButtonListener implements ActionListener {

        @Override

        public void actionPerformed(ActionEvent e) {

            String firstNumberStr = firstNumberField.getText();

            String secondNumberStr = secondNumberField.getText();

            try {

                int firstNumber = Integer.parseInt(firstNumberStr);

                int secondNumber = Integer.parseInt(secondNumberStr);

                int sum = firstNumber + secondNumber;

                // Update the result label

                resultLabel.setText(String.valueOf(sum));

            } catch (NumberFormatException ex) {

                JOptionPane.showMessageDialog(frame, "Invalid number format", "Error", JOptionPane.ERROR\_MESSAGE);

            }

        }

    }

    private class ClearButtonListener implements ActionListener {

        @Override

        public void actionPerformed(ActionEvent e) {

            // Clear the text fields and result label

            firstNumberField.setText("");

            secondNumberField.setText("");

            resultLabel.setText("");

        }

    }

    private class ExitButtonListener implements ActionListener {

        @Override

        public void actionPerformed(ActionEvent e) {

            // Exit the application

            System.exit(0);

        }

    }

    public static void main(String[] args) {

              new NumberAddition();

    }

}

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 multipleTable {

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);

        System.out.print("Enter the number:");

        int n=sc.nextInt();

        for(int i=1;i<=10;i++){

            System.out.println(n+"\*"+i+"="+n\*i);

        }

    }

}

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

import java.util.\*;

public class checkPrimeOrNot {

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter Number for checking:");

        int n=sc.nextInt();

        boolean prime=true;

      for(int i=2;i<n;i++){

        if(n%i==0){

           prime=false;

           break;

        }

      }

      if(prime){

        System.out.println(n+" is prime number.");

      }

      else{

        System.out.println(n+" is not prime number.");

      }

    }

}

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 DiamondPattern {

    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();

        scanner.close();

        // Upper half of the diamond

        for (int i = 1; i <= rows; i++) {

            for (int j = 1; j <= rows - i; j++) {

                System.out.print(" ");

            }

            for (int k = 1; k <= 2 \* i - 1; k++) {

                System.out.print("\*");

            }

            System.out.println();

        }

        // Lower half of the diamond

        for (int i = rows - 1; i >= 1; i--) {

            for (int j = 1; j <= rows - i; j++) {

                System.out.print(" ");

            }

            for (int k = 1; k <= 2 \* i - 1; k++) {

                System.out.print("\*");

            }

            System.out.println();

        }

    }

}

13.

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

import java.util.Scanner;

public class transposeOfMatrix {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Read the dimensions of the matrix

        System.out.print("Enter the number of rows in the matrix: ");

        int rows = scanner.nextInt();

        System.out.print("Enter the number of columns in the matrix: ");

        int columns = scanner.nextInt();

        // Create the matrix

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

        // Read the matrix elements

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

        for (int i = 0; i < rows; i++) {

            for (int j = 0; j < columns; j++) {

                matrix[i][j] = scanner.nextInt();

            }

        }

        System.out.println("Entered matrix is: ");

        for (int i = 0; i <rows; i++) {

            for (int j = 0; j <columns; j++) {

                System.out.print(matrix[i][j] + " ");

            }

            System.out.println();

        }

        // Transpose the matrix

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

        for (int i = 0; i < columns; i++) {

            for (int j = 0; j < rows; j++) {

                transposeMatrix[i][j] = matrix[j][i];

            }

        }

        // Print the transpose matrix

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

        for (int i = 0; i < columns; i++) {

            for (int j = 0; j < rows; j++) {

                System.out.print(transposeMatrix[i][j] + " ");

            }

            System.out.println();

        }

        scanner.close();

    }

}

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.io.FileReader;

import java.io.IOException;

import java.util.Scanner;

public class fileCountWords{

    public static void main(String[] args) {

        String filename = "C:/Users/sahil/OneDrive/Desktop/Sahil.txt";

        try {

            int wordCount = countWords(filename);

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

        } catch (FileNotFoundException e) {

            System.out.println("File not found: " + filename);

        }

    }

    public static int countWords(String filename) throws FileNotFoundException {

        int count = 0;

        File file = new File(filename);

        FileReader file1=new FileReader(filename);

        try{

            int i;

            while((i=file1.read())!=-1){

                System.out.print((char)i);

            }

            System.out.println();

        }

        catch(IOException e){

            System.out.println("File not found.");

        }

        Scanner scanner = new Scanner(file);

        while (scanner.hasNext()) {

            scanner.next();

            count++;

        }

        scanner.close();

        return count;

    }

}

14. 

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

import java.util.ArrayList;

import java.util.\*;;

public class IterateArrayList {

    public static void main(String[] args) {

        ArrayList<Integer> list=new ArrayList<Integer>();

        list.add(20);

        list.add(30);

        list.add(40);

        Iterator it=list.iterator();

        System.out.println("Using Iterator: ");

        while(it.hasNext()){

            System.out.println(it.next()+" ");

        }

        System.out.println("Using Advanaced for loop: ");

        for(int i:list){

            System.out.println(i+" ");

        }

        System.out.println("Using for loop: ");

        for(int i=0;i<list.size();i++){

            System.out.println(list.get(i));

        }

    }

}

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 countWords {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Read the input string

        System.out.print("Enter the string: ");

        String input = scanner.nextLine();

        // Split the string into words

        String[] words = input.split(" ");

        // Create a HashMap to store word counts

        Map<String, Integer> wordCountMap = new HashMap<>();

        // Count the occurrences of each word

        for (String word : words) {

            if (wordCountMap.containsKey(word)) {

                // Increment the count if the word already exists in the map

                int count = wordCountMap.get(word);

                wordCountMap.put(word, count + 1);

            } else {

                // Add the word to the map with a count of 1

                wordCountMap.put(word, 1);

            }

        }

        // Print the word count map

        System.out.println(wordCountMap);

        scanner.close();

    }

}

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);

        // Read 10 strings from console

        String[] strings = new String[10];

        System.out.println("Enter 10 strings:");

        for (int i = 0; i < 10; i++) {

            strings[i] = scanner.nextLine();

        }

        // Sort the strings

        Arrays.sort(strings);

        // Print the sorted strings

        System.out.println("Sorted strings:");

        for (String string : strings) {

            System.out.println(string);

        }

        // Combine two strings

        System.out.print("Enter the first string: ");

        String firstString = scanner.nextLine();

        System.out.print("Enter the second string: ");

        String secondString = scanner.nextLine();

        String combinedString = firstString + secondString;

        System.out.println("Combined string: " + combinedString);

        // Reverse the first string and display it

        String reversedString = reverseString(firstString);

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

        scanner.close();

    }

    public static String reverseString(String str) {

        char temp;

        String ans="";

        for(int i=0;i<str.length();i++){

            temp=str.charAt(i);

            ans=temp+ans;

        }

        return ans;

    }

}

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.ArrayList;

import java.util.Scanner;

class Person {

    String name;

    int age;

    Person(String name, int age) {

        this.name = name;

        this.age = age;

    }

}

class Employee extends Person {

    String designation;

    double salary;

    Employee(String name, int age, String designation, double salary) {

        super(name, age);

        this.designation = designation;

        this.salary = salary;

    }

}

public class EmplyeeSalary{

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        ArrayList<Employee> employees = new ArrayList<>();

        // Accept data for 5 persons and store them in a list of Employee objects

        for (int i = 0; i < 5; i++) {

            System.out.print("Enter name: ");

            String name = scanner.nextLine();

            System.out.print("Enter age: ");

            int age = Integer.parseInt(scanner.nextLine());

            System.out.print("Enter designation: ");

            String designation = scanner.nextLine();

            System.out.print("Enter salary: ");

            double salary = Double.parseDouble(scanner.nextLine());

            Employee employee = new Employee(name, age, designation, salary);

            employees.add(employee);

        }

        // Display the name of employees with a salary greater than 5000

        System.out.println("Employees with a salary greater than 5000:");

        for (Employee employee : employees) {

            if (employee.salary > 5000) {

                System.out.println(employee.name);

            }

        }

    }

}

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.

interface Account {

    void set(String name, int accountNumber, double balance);

    void display();

}

interface Person {

    void store(String name);

    void disp();

}

class Customer implements Account, Person {

    private String name;

    private int accountNumber;

    private double balance;

    @Override

    public void set(String name, int accountNumber, double balance) {

        this.name = name;

        this.accountNumber = accountNumber;

        this.balance = balance;

    }

    @Override

    public void display() {

        System.out.println("Name: " + name);

        System.out.println("Account number: " + accountNumber);

        System.out.println("Balance: " + balance);

    }

    @Override

    public void store(String name) {

        this.name = name;

    }

    @Override

    public void disp() {

        System.out.println("Name: " + name);

    }

    public double calculateInterest() {

        // calculate interest

        double interest = balance \* 0.05;

        return interest;

    }

}

public class MultipleInterface {

    public static void main(String[] args) {

        Customer customer = new Customer();

        customer.store("Sahil Mandhare");

        customer.set("Sahil Mandhare", 123456, 10000);

        customer.display();

        System.out.println("Interest: " + customer.calculateInterest());

        customer.disp();

    }

}

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.  
"



class VideoTape {

    protected String title;

    protected int length;

    protected boolean available;

    public VideoTape(String title, int length, boolean available) {

        this.title = title;

        this.length = length;

        this.available = available;

    }

    public void show() {

        System.out.println("Title: " + title);

        System.out.println("Length: " + length + " minutes");

        System.out.println("Available: " + available);

    }

}

class Movie extends VideoTape {

    private String director;

    private int rating;

    public Movie(String title, int length, boolean available, String director, int rating) {

        super(title, length, available);

        this.director = director;

        this.rating = rating;

    }

    @Override

    public void show() {

        super.show();

        System.out.println("Director: " + director);

        System.out.println("Rating: " + rating);

    }

}

class MusicVideo extends VideoTape {

    private String artist;

    private String category;

    public MusicVideo(String title, int length, boolean available, String artist, String category) {

        super(title, length, available);

        this.artist = artist;

        this.category = category;

    }

    @Override

    public void show() {

        super.show();

        System.out.println("Artist: " + artist);

        System.out.println("Category: " + category);

    }

}

public class HerichiInheritance {

    public static void main(String[] args) {

        VideoTape videoTape = new VideoTape("Generic Video", 120, true);

        videoTape.show();

        System.out.println();

        Movie movie = new Movie("The Avengers", 180, true, "Joss Whedon", 9);

        movie.show();

        System.out.println();

        MusicVideo musicVideo = new MusicVideo("Shape of You", 4, true, "Ed Sheeran", "Pop");

        musicVideo.show();

    }

}

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;

class Student {

    private String name;

    private int grade;

    private ArrayList<String> courses;

    public Student(String name, int grade) {

        this.name = name;

        this.grade = grade;

        this.courses = new ArrayList<>();

    }

    public void addCourse(String course) {

        courses.add(course);

    }

    public void removeCourse(String course) {

        courses.remove(course);

    }

    public void displayCourses() {

        System.out.println("Courses for " + name + ":");

        for (String course : courses) {

            System.out.println(course);

        }

    }

    public String getName() {

        return name;

    }

    public int getGrade() {

        return grade;

    }

}

public class studentCourse {

    public static void main(String[] args) {

        Student student = new Student("Sahil Mandhare", 10);

        student.addCourse("Math");

        student.addCourse("Science");

        student.addCourse("English");

        student.displayCourses();

        student.removeCourse("Science");

        System.out.println("After removing Science course:");

        student.displayCourses();

    }

}

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.

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 int employeeId;

    private String jobTitle;

    public Employee(String firstName, String lastName, int employeeId, String jobTitle) {

        super(firstName, lastName);

        this.employeeId = employeeId;

        this.jobTitle = jobTitle;

    }

    public int getEmployeeId() {

        return employeeId;

    }

    @Override

    public String getLastName() {

        return super.getLastName() + " (" + jobTitle + ")";

    }

}

public class jobTitle {

    public static void main(String[] args) {

        Person person = new Person("Sahil", "Mandhare");

        System.out.println("Person:");

        System.out.println("First Name: " + person.getFirstName());

        System.out.println("Last Name: " + person.getLastName());

        System.out.println();

        Employee employee = new Employee("Sahil", "Deshmukh", 12110788, "Manager");

        System.out.println("Employee:");

        System.out.println("First Name: " + employee.getFirstName());

        System.out.println("Last Name: " + employee.getLastName());

        System.out.println("Employee ID: " + employee.getEmployeeId());

    }

}

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.

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.

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 valid range (15-21).");

        }

        if (!name.matches("[a-zA-Z]+")) {

            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 userDefinedExce{

    public static void main(String[] args) {

        try {

            Student student1 = new Student(1, "SahilMandhare", 20, "ENTC");

            student1.displayInfo();

        } catch (AgeNotWithinRangeException e) {

            System.out.println("Exception: " + e.getMessage());

        } catch (NameNotValidException e) {

            System.out.println("Exception: " + e.getMessage());

        }

        System.out.println();

        try {

            Student student2 = new Student(2, "Jane", 33, "Physics");

            student2.displayInfo();

        } catch (AgeNotWithinRangeException e) {

            System.out.println("Exception: " + e.getMessage());

        } catch (NameNotValidException e) {

            System.out.println("Exception: " + e.getMessage());

        }

    }

}

25. 