1.

1.1) Program to remove all repeated elements from an array

package programs;

//1.1) Program to remove all repeated elements from an array

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.Scanner;

public class Lab1 {

    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[] array = new int[n];

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

        for (int i = 0; i < n; i++) {

            array[i] = scanner.nextInt();

        }

        int[] uniqueArray = removeDuplicates(array);

        System.out.println("Original Array: " + Arrays.toString(array));

        System.out.println("Array without Duplicates: " + Arrays.toString(uniqueArray));

    }

    public static int[] removeDuplicates(int[] array) {

        List<Integer> uniqueList = new ArrayList<>();

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

            if (!uniqueList.contains(array[i])) {

                uniqueList.add(array[i]);

            }

        }

        int[] uniqueArray = new int[uniqueList.size()];

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

            uniqueArray[i] = uniqueList.get(i);

        }

        return uniqueArray;

    }

}

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

package programs;

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

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.Scanner;

public class Lab1ka2 {

    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[] array1 = new int[n1];

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

        for (int i = 0; i < n1; i++) {

            array1[i] = scanner.nextInt();

        }

        System.out.print("Enter the number of elements in the second array: ");

        int n2 = scanner.nextInt();

        int[] array2 = new int[n2];

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

        for (int i = 0; i < n2; i++) {

            array2[i] = scanner.nextInt();

        }

        int[] commonElements = findCommonElements(array1, array2);

        System.out.println("Array 1: " + Arrays.toString(array1));

        System.out.println("Array 2: " + Arrays.toString(array2));

        System.out.println("Common Elements: " + Arrays.toString(commonElements));

    }

    public static int[] findCommonElements(int[] array1, int[] array2) {

        List<Integer> commonList = new ArrayList<>();

        for (int num1 : array1) {

            for (int num2 : array2) {

                if (num1 == num2 && !commonList.contains(num1)) {

                    commonList.add(num1);

                }

            }

        }

        int[] commonElements = new int[commonList.size()];

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

            commonElements[i] = commonList.get(i);

        }

        return commonElements;

    }

}

2.

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

package programs;

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

import java.util.HashMap;

import java.util.Map;

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();

        Map<String, Integer> wordCountMap = countDuplicateWords(inputString);

        System.out.println("Duplicate Words Count:");

        for (Map.Entry<String, Integer> entry : wordCountMap.entrySet()) {

            if (entry.getValue() > 1) {

                System.out.println(entry.getKey() + ": " + entry.getValue());

            }

        }

    }

    public static Map<String, Integer> countDuplicateWords(String inputString) {

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

        // Split the input string into words

        String[] words = inputString.toLowerCase().split(" ");

        // Count the occurrence of each word

        for (String word : words) {

            if (wordCountMap.containsKey(word)) {

                wordCountMap.put(word, wordCountMap.get(word) + 1);

            } else {

                wordCountMap.put(word, 1);

            }

        }

        return wordCountMap;

    }

}

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

package programs;

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

public class StringContainsExample {

    public static void main(String[] args) {

        String word = "umbrella";

        boolean containsE = checkIfContainsE(word);

        if (containsE) {

            System.out.println("The string '" + word + "' contains 'e'.");

        } else {

            System.out.println("The string '" + word + "' does not contain 'e'.");

        }

    }

    public static boolean checkIfContainsE(String word) {

        return word.contains("e");

    }

}

/\*

 \* public class StringContainsExample {

 \* public static void main(String[] args) {

 \* String word = "umbrella";

 \*

 \* boolean containsE = checkIfContainsE(word);

 \*

 \* if (containsE) {

 \* System.out.println("The string '" + word + "' contains 'e'.");

 \* } else {

 \* System.out.println("The string '" + word + "' does not contain 'e'.");

 \* }

 \* }

 \*

 \* public static boolean checkIfContainsE(String word) {

 \* for (int i = 0; i < word.length(); i++) {

 \* if (word.charAt(i) == 'e') {

 \* return true;

 \* }

 \* }

 \* return false;

 \* }

 \* }

 \*/

3.

3.1)Java Program to Reverse a String.

package programs;

//3.1)Java Program to Reverse a String.

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);

    }

    public static String reverseString(String inputString) {

        StringBuilder reversed = new StringBuilder();

        // Iterate through the characters of the inputString in reverse order

        for (int i = inputString.length() - 1; i >= 0; i--) {

            reversed.append(inputString.charAt(i));

        }

        return reversed.toString();

    }

}

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

package programs;

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

import java.util.Scanner;

public class PalindromeChecker {

    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.");

        }

    }

    public static boolean checkPalindrome(String inputString) {

        String reversedString = reverseString(inputString);

        return inputString.equalsIgnoreCase(reversedString);

    }

    public static String reverseString(String inputString) {

        StringBuilder reversed = new StringBuilder();

        for (int i = inputString.length() - 1; i >= 0; i--) {

            reversed.append(inputString.charAt(i));

        }

        return reversed.toString();

    }

}

/\*

 \* import java.util.Scanner;

 \*

 \* public class PalindromeChecker {

 \* 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.");

 \* }

 \* }

 \*

 \* public static boolean checkPalindrome(String inputString) {

 \* int left = 0;

 \* int right = inputString.length() - 1;

 \*

 \* while (left < right) {

 \* if (inputString.charAt(left) != inputString.charAt(right)) {

 \* return false;

 \* }

 \* left++;

 \* right--;

 \* }

 \*

 \* return true;

 \* }

 \* }

 \*/

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.

package programs;

/\*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;

abstract class Vehicle {

    public abstract void move();

}

class Car extends Vehicle {

    public void move() {

        System.out.println("The car moves on the road.");

    }

}

class Helicopter extends Vehicle {

    public void move() {

        System.out.println("The helicopter moves in the air.");

    }

}

class Train extends Vehicle {

    public void move() {

        System.out.println("The train moves on the tracks.");

    }

}

public class VehicleFactory {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the type of vehicle you want to order (car/helicopter/train): ");

        String vehicleType = scanner.nextLine().toLowerCase(); // Convert input to lowercase

        Vehicle vehicle = null;

        switch (vehicleType) {

            case "car":

                vehicle = new Car();

                break;

            case "helicopter":

                vehicle = new Helicopter();

                break;

            case "train":

                vehicle = new Train();

                break;

            default:

                System.out.println("Invalid vehicle type entered.");

                break;

        }

        if (vehicle != null) {

            vehicle.move();

        }

    }

}

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.

package programs;

/\*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. \*/

import java.util.Scanner;

abstract class Marks {

    public abstract double getPercentage();

}

class A extends Marks {

    private double subject1;

    private double subject2;

    private double subject3;

    public A(double subject1, double subject2, double subject3) {

        this.subject1 = subject1;

        this.subject2 = subject2;

        this.subject3 = subject3;

    }

    @Override

    public double getPercentage() {

        double totalMarks = subject1 + subject2 + subject3;

        return (totalMarks / 300) \* 100;

    }

}

class B extends Marks {

    private double subject1;

    private double subject2;

    private double subject3;

    private double subject4;

    public B(double subject1, double subject2, double subject3, double subject4) {

        this.subject1 = subject1;

        this.subject2 = subject2;

        this.subject3 = subject3;

        this.subject4 = subject4;

    }

    @Override

    public double getPercentage() {

        double totalMarks = subject1 + subject2 + subject3 + subject4;

        return (totalMarks / 400) \* 100;

    }

}

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the marks for Student A:");

        System.out.print("Subject 1: ");

        double subject1A = scanner.nextDouble();

        System.out.print("Subject 2: ");

        double subject2A = scanner.nextDouble();

        System.out.print("Subject 3: ");

        double subject3A = scanner.nextDouble();

        System.out.println("Enter the marks for Student B:");

        System.out.print("Subject 1: ");

        double subject1B = scanner.nextDouble();

        System.out.print("Subject 2: ");

        double subject2B = scanner.nextDouble();

        System.out.print("Subject 3: ");

        double subject3B = scanner.nextDouble();

        System.out.print("Subject 4: ");

        double subject4B = scanner.nextDouble();

        A studentA = new A(subject1A, subject2A, subject3A);

        B studentB = new B(subject1B, subject2B, subject3B, subject4B);

        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**

package programs;

/\*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

 \*/

import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.math.\*;

import java.util.regex.\*;

//Write your code here

class Arithmetic {

    public int add(int a, int b) {

        int sum = a + b;

        return sum;

    }

}

class Adder extends Arithmetic {

    public int callAdd(int a, int b) {

        return add(a, b);

    }

}

public class ArithmeticAdder {

    public static void main(String[] args) {

        // Create a new Adder object

        Adder a = new Adder();

        // Print the name of the superclass on a new line

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

        // Print the result of 3 calls to Adder's `add(int,int)` method as 3

        // space-separated integers:

        System.out.print(a.add(10, 32) + " " + a.add(10, 3) + " " + a.add(10, 10) + "\n");

    }

}

**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”.

package programs;

/\*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 Calculator {

    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(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.

package programs;

/\*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.Scanner;

public class PhoneBook {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter the number of entries in the phone book

        System.out.print("Enter the number of entries in the phone book: ");

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

        // Create a HashMap to store the phone book entries

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

        // Read and store 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);

        }

        // Read the queries and print the corresponding phone numbers

        System.out.println("\nEnter the queries (enter 'q' to quit):");

        while (true) {

            System.out.print("Query: ");

            String query = scanner.nextLine();

            if (query.equals("q")) {

                break;

            }

            if (phoneBook.containsKey(query)) {

                String phoneNumber = phoneBook.get(query);

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

            } else {

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

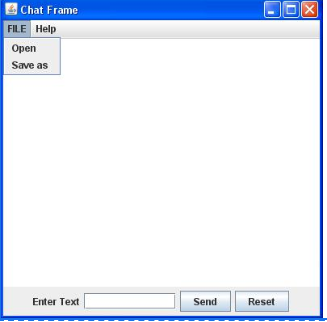
            }

        }

        scanner.close();

    }

}

9. 

package programs;

//8. menu file open save as ---help ---below enter text field ---send btn ---reset btn--

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. 

package programs;

//Additon of 1st no and 2nd number and display result btn add and clear and exit

import java.awt.EventQueue;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class NumberAddition {

    private JFrame frame;

    private JTextField f1;

    private JTextField s1;

    private JTextField r1;

    /\*\*

     \* Launch the application.

     \*/

    public static void main(String[] args) {

        EventQueue.invokeLater(new Runnable() {

            public void run() {

                try {

                    NumberAddition window = new NumberAddition();

                    window.frame.setVisible(true);

                } catch (Exception e) {

                    e.printStackTrace();

                }

            }

        });

    }

    /\*\*

     \* Create the application.

     \*/

    public NumberAddition() {

        initialize();

    }

    /\*\*

     \* Initialize the contents of the frame.

     \*/

    private void initialize() {

        frame = new JFrame();

        frame.setBounds(100, 100, 450, 300);

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

        frame.getContentPane().setLayout(null);

        JLabel label1 = new JLabel("Number Addition");

        label1.setForeground(Color.BLUE);

        label1.setBounds(20, 10, 100, 30);

        frame.getContentPane().add(label1);

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

        first.setBounds(10, 40, 120, 20);

        frame.getContentPane().add(first);

        f1 = new JTextField();

        f1.setBounds(140, 40, 140, 20);

        frame.getContentPane().add(f1);

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

        second.setBounds(10, 65, 120, 20);

        frame.getContentPane().add(second);

        s1 = new JTextField();

        s1.setBounds(140, 65, 140, 20);

        frame.getContentPane().add(s1);

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

        result.setBounds(10, 90, 120, 20);

        frame.getContentPane().add(result);

        r1 = new JTextField();

        r1.setBounds(140, 90, 140, 20);

        frame.getContentPane().add(r1);

        JButton add = new JButton("ADD");

        add.setBounds(90, 120, 80, 20);

        frame.getContentPane().add(add);

        // Add ActionListener to perform addition

        add.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                String num1 = f1.getText();

                String num2 = s1.getText();

                if (!num1.isEmpty() && !num2.isEmpty()) {

                    int resultVal = Integer.parseInt(num1) + Integer.parseInt(num2);

                    r1.setText(Integer.toString(resultVal));

                } else {

                    JOptionPane.showMessageDialog(frame, "Please enter both numbers.");

                }

            }

        });

        JButton clear = new JButton("CLEAR");

        clear.setBounds(200, 120, 80, 20);

        frame.getContentPane().add(clear);

        // Add ActionListener to clear the fields

        clear.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                f1.setText("");

                s1.setText("");

                r1.setText("");

            }

        });

        JButton exit = new JButton("Exit");

        exit.setBounds(250, 200, 80, 20);

        frame.add(exit);

        // Add ActionListener to exit the program

        exit.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                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

package programs;

/\*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("Enter a number: ");

        int number = scanner.nextInt();

        System.out.println("Multiplication Table for " + number + ":");

        for (int i = 1; i <= 10; i++) {

            int product = number \* i;

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

        }

        scanner.close();

    }

}

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

package programs;

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

import java.util.Scanner;

public class PrimeExample {

    public static void main(String args[]) {

        Scanner scanner = new Scanner(System.in);

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

        int n = scanner.nextInt();

        int m = n / 2;

        int flag = 0;

        if (n == 0 || n == 1) {

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

        } else {

            for (int i = 2; i <= m; i++) {

                if (n % i == 0) {

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

                    flag = 1;

                    break;

                }

            }

            if (flag == 0) {

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

            }

        }

        scanner.close();

    }

}

12. Write a Java program to display the pattern like a diamond.  
Input number of rows (half of the diamond) :7 Expected Output :  
  
  
\*   
\*\*\*   
\*\*\*\*\*   
\*\*\*\*\*\*\*   
\*\*\*\*\*\*\*\*\*   
\*\*\*\*\*\*\*\*\*\*\*   
\*\*\*\*\*\*\*\*\*\*\*\*\*   
\*\*\*\*\*\*\*\*\*\*\*   
\*\*\*\*\*\*\*\*\*   
\*\*\*\*\*\*\*   
\*\*\*\*\*   
\*\*\*   
\*

package programs;

/\*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 HalfDiamond {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the 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) Write Java Program to find the transpose of a given matrix .

package programs;

//13.1) Write Java 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 in the matrix: ");

        int rows = scanner.nextInt();

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

        int columns = scanner.nextInt();

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

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

        // Read the matrix elements from the user

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

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

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

            }

        }

        // Transpose the matrix

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

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

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

                transpose[j][i] = matrix[i][j];

            }

        }

        // Print the transpose matrix

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

        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();

    }

}

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

package programs;

//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 {

            // Open the text file

            File file = new File("D:/avitsem4/oop/lab \_exam/Labexam/src/programs/Labexamquestions.txt");

            Scanner scanner = new Scanner(file);

            int wordCount = 0;

            // Iterate through each line of the file

            while (scanner.hasNextLine()) {

                String line = scanner.nextLine();

                // Split the line into words using whitespace as the delimiter

                String[] words = line.split("\\s+");

                // Update the word count

                wordCount += words.length;

            }

            // Close the scanner

            scanner.close();

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

        } catch (FileNotFoundException e) {

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

        }

    }

}

14. 

package programs;

import javax.script.ScriptEngine;

import javax.script.ScriptEngineManager;

import javax.script.ScriptException;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class SwingCalculator extends JFrame {

    private JTextField textField;

    private JPanel buttonPanel;

    public SwingCalculator() {

        setTitle("Swing Calculator");

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setLayout(new BorderLayout());

        // Text field at the top

        textField = new JTextField();

        add(textField, BorderLayout.NORTH);

        // Button panel in the center

        buttonPanel = new JPanel();

        buttonPanel.setLayout(new GridLayout(4, 4));

        add(buttonPanel, BorderLayout.CENTER);

        // Add buttons to the button panel

        addButton("7");

        addButton("8");

        addButton("9");

        addButton("\*");

        addButton("4");

        addButton("5");

        addButton("6");

        addButton("-");

        addButton("1");

        addButton("2");

        addButton("3");

        addButton("+");

        addButton("C");

        addButton("0");

        addButton("=");

        addButton("/");

        pack();

        setLocationRelativeTo(null);

    }

    private void addButton(String label) {

        JButton button = new JButton(label);

        button.addActionListener(new ButtonClickListener());

        buttonPanel.add(button);

    }

    private class ButtonClickListener implements ActionListener {

        public void actionPerformed(ActionEvent event) {

            String command = event.getActionCommand();

            if (command.equals("=")) {

                String expression = textField.getText();

                double result = calculateExpression(expression);

                textField.setText(Double.toString(result));

            } else if (command.equals("C")) {

                textField.setText("");

            } else {

                textField.setText(textField.getText() + command);

            }

        }

        private double calculateExpression(String expression) {

            ScriptEngineManager manager = new ScriptEngineManager();

            ScriptEngine engine = manager.getEngineByName("JavaScript");

            try {

                Object result = engine.eval(expression);

                if (result instanceof Integer) {

                    return (Integer) result;

                } else if (result instanceof Double) {

                    return (Double) result;

                }

            } catch (ScriptException e) {

                JOptionPane.showMessageDialog(null, "Invalid Expression", "Error", JOptionPane.ERROR\_MESSAGE);

            }

            return 0.0;

        }

    }

    public static void main(String[] args) {

        SwingCalculator calculator = new SwingCalculator();

        calculator.setVisible(true);

    }

}

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

package programs;

/\*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.Iterator;

import java.util.Scanner;

public class ArrayListIteration {

    public static void main(String[] args) {

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

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter three numbers:");

        for (int i = 0; i < 3; i++) {

            int num = scanner.nextInt();

            numbers.add(num);

        }

        System.out.println("ArrayList elements:");

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

        iterateUsingIterator(numbers);

        System.out.println("Advanced For Loop:");

        iterateUsingAdvancedForLoop(numbers);

        System.out.println("For Loop:");

        iterateUsingForLoop(numbers);

        scanner.close();

    }

    // Iterate using Iterator

    public static void iterateUsingIterator(ArrayList<Integer> numbers) {

        Iterator<Integer> iterator = numbers.iterator();

        while (iterator.hasNext()) {

            System.out.println(iterator.next());

        }

    }

    // Iterate using Advanced For Loop

    public static void iterateUsingAdvancedForLoop(ArrayList<Integer> numbers) {

        for (Integer number : numbers) {

            System.out.println(number);

        }

    }

    // Iterate using regular For Loop

    public static void iterateUsingForLoop(ArrayList<Integer> numbers) {

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

            System.out.println(numbers.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}

package programs;

/\*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.Scanner;

public class StringWordCount {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter a string:");

        String input = scanner.nextLine();

        HashMap<String, Integer> wordCountMap = countWords(input);

        System.out.println(wordCountMap);

        scanner.close();

    }

    public static HashMap<String, Integer> countWords(String input) {

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

        // Split the input string into words

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

        // Count the occurrence of each word

        for (String word : words) {

            if (wordCountMap.containsKey(word)) {

                int count = wordCountMap.get(word);

                wordCountMap.put(word, count + 1);

            } else {

                wordCountMap.put(word, 1);

            }

        }

        return wordCountMap;

    }

}

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 .

package programs;

/\*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];

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

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

            strings[i] = scanner.nextLine();

        }

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

        sortStrings(strings);

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

        String combinedString = combineStrings(strings);

        System.out.println(combinedString);

        System.out.println("Reversed first string:");

        String reversedString = reverseString(strings[0]);

        System.out.println(reversedString);

        scanner.close();

    }

    public static void sortStrings(String[] strings) {

        Arrays.sort(strings);

        for (String string : strings) {

            System.out.println(string);

        }

    }

    public static String combineStrings(String[] strings) {

        StringBuilder combined = new StringBuilder();

        for (String string : strings) {

            combined.append(string);

        }

        return combined.toString();

    }

    public static String reverseString(String string) {

        StringBuilder reversed = new StringBuilder(string);

        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

package programs;

/\*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 accept() {

        Scanner scanner = new Scanner(System.in);

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

        name = scanner.nextLine();

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

        age = scanner.nextInt();

    }

}

class Employee extends Mainm {

    private String designation;

    private double salary;

    public void accept() {

        super.accept();

        Scanner scanner = new Scanner(System.in);

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

        designation = scanner.nextLine();

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

        salary = scanner.nextDouble();

    }

    public void display() {

        if (salary > 5000) {

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

        }

    }

}

public class EmployeeIN {

    public static void main(String[] args) {

        Employee[] employees = new Employee[5];

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter data for 5 employees:");

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

            System.out.println("Employee " + (i + 1));

            employees[i] = new Employee();

            employees[i].accept();

        }

        System.out.println("\nEmployees with salary greater than 5000:");

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

            employees[i].display();

        }

    }

}

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.

package programs;

/\*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;

// Account interface

interface Account {

    void set(String accountNumber, double balance);

    void display();

}

// Person interface

interface Perso {

    void store(String name);

    void disp();

}

// Customer class implementing Account and Person interfaces

class Customer implements Account, Perso {

    private String name;

    private String accountNumber;

    private double balance;

    @Override

    public void set(String accountNumber, double balance) {

        this.accountNumber = accountNumber;

        this.balance = balance;

    }

    @Override

    public void display() {

        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 void calculateInterest(double interestRate) {

        double interest = balance \* interestRate / 100;

        balance += interest;

        System.out.println("Interest: $" + interest);

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

    }

}

// Main class

public class MultipleInheritanceExample {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        Customer customer = new Customer();

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

        String name = scanner.nextLine();

        customer.store(name);

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

        String accountNumber = scanner.nextLine();

        System.out.print("Enter balance: $");

        double balance = scanner.nextDouble();

        scanner.nextLine();

        customer.set(accountNumber, balance);

        System.out.println("\nCustomer Information:");

        customer.disp();

        customer.display();

        System.out.print("\nEnter interest rate (%): ");

        double interestRate = scanner.nextDouble();

        System.out.println("\nAccount Information with Interest:");

        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.  
"



package programs;

/\*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. \*/

import java.util.Scanner;

class VideoTape {

    private String title;

    private int length;

    public VideoTape(String title, int length) {

        this.title = title;

        this.length = length;

    }

    public String getTitle() {

        return title;

    }

    public int getLength() {

        return length;

    }

    @Override

    public String toString() {

        return "VideoTape: " + title + " (" + length + " minutes)";

    }

}

class Movie extends VideoTape {

    private String rating;

    public Movie(String title, int length, String rating) {

        super(title, length);

        this.rating = rating;

    }

    public String getRating() {

        return rating;

    }

    @Override

    public String toString() {

        return "Movie: " + getTitle() + " (" + getLength() + " minutes, rated " + rating + ")";

    }

}

class MusicVideo extends VideoTape {

    private String artist;

    public MusicVideo(String title, int length, String artist) {

        super(title, length);

        this.artist = artist;

    }

    public String getArtist() {

        return artist;

    }

    @Override

    public String toString() {

        return "MusicVideo: " + getTitle() + " (" + getLength() + " minutes, by " + artist + ")";

    }

}

public class VideoTapeM {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the title of the video tape: ");

        String title = scanner.nextLine();

        System.out.print("Enter the length of the video tape (in minutes): ");

        int length = scanner.nextInt();

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

        System.out.print("Enter the rating of the movie (if applicable): ");

        String rating = scanner.nextLine();

        System.out.print("Enter the artist of the music video (if applicable): ");

        String artist = scanner.nextLine();

        scanner.close();

        VideoTape tape1 = new VideoTape(title, length);

        System.out.println(tape1);

        if (!rating.isEmpty()) {

            Movie movie1 = new Movie(title, length, rating);

            System.out.println(movie1);

        }

        if (!artist.isEmpty()) {

            MusicVideo musicVideo1 = new MusicVideo(title, length, artist);

            System.out.println(musicVideo1);

        }

    }

}

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.

package programs;

/\*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 int grade;

    private List<String> courses;

    public Student(String name, int grade) {

        this.name = name;

        this.grade = grade;

        this.courses = new ArrayList<>();

    }

    public String getName() {

        return name;

    }

    public int getGrade() {

        return grade;

    }

    public List<String> getCourses() {

        return courses;

    }

    public void addCourse(String course) {

        courses.add(course);

    }

    public void removeCourse(String course) {

        courses.remove(course);

    }

}

public class AddRemoveCourses {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter student's name: ");

        String name = scanner.nextLine();

        System.out.print("Enter student's grade: ");

        int grade = scanner.nextInt();

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

        // Create a new student

        Student student = new Student(name, grade);

        // Prompt for adding courses

        String course;

        char choice;

        do {

            System.out.print("Enter course name to add: ");

            course = scanner.nextLine();

            student.addCourse(course);

            System.out.print("Do you want to add another course? (Y/N): ");

            choice = scanner.nextLine().charAt(0);

        } while (choice == 'Y' || choice == 'y');

        // Prompt for removing courses

        do {

            System.out.print("Enter course name to remove: ");

            course = scanner.nextLine();

            student.removeCourse(course);

            System.out.print("Do you want to remove another course? (Y/N): ");

            choice = scanner.nextLine().charAt(0);

        } while (choice == 'Y' || choice == 'y');

        scanner.close();

        // Display student details

        System.out.println("Name: " + student.getName());

        System.out.println("Grade: " + student.getGrade());

        System.out.println("Courses: " + student.getCourses());

    }

}

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.

package programs;

/\*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.\*/

import java.util.Scanner;

class Person1 {

    private String firstName;

    private String lastName;

    public Person1(String firstName, String lastName) {

        this.firstName = firstName;

        this.lastName = lastName;

    }

    public String getFirstName() {

        return firstName;

    }

    public String getLastName() {

        return lastName;

    }

}

class Employee1 extends Person1 {

    private int employeeId;

    private String jobTitle;

    public Employee1(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 Employeeperson {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Input for Person

        System.out.print("Enter Person's First Name: ");

        String personFirstName = scanner.nextLine();

        System.out.print("Enter Person's Last Name: ");

        String personLastName = scanner.nextLine();

        // Input for Employee

        System.out.print("Enter Employee's First Name: ");

        String employeeFirstName = scanner.nextLine();

        System.out.print("Enter Employee's Last Name: ");

        String employeeLastName = scanner.nextLine();

        System.out.print("Enter Employee's ID: ");

        int employeeId = scanner.nextInt();

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

        System.out.print("Enter Employee's Job Title: ");

        String jobTitle = scanner.nextLine();

        // Create Person and Employee objects

        Person1 person = new Person1(personFirstName, personLastName);

        Employee1 employee = new Employee1(employeeFirstName, employeeLastName, employeeId, jobTitle);

        System.out.println("\nPerson:");

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

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

        System.out.println("\nEmployee:");

        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.

package programs;

/\*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.io.\*;

import java.util.\*;

class GFG {

    static int findLongestConseqSubseq(int arr[], int n) {

        // Sort the array

        Arrays.sort(arr);

        int ans = 0, count = 0;

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

        v.add(arr[0]);

        // Insert repeated elements

        // only once in the vector

        for (int i = 1; i < n; i++) {

            if (arr[i] != arr[i - 1])

                v.add(arr[i]);

        }

        // Find the maximum length

        // by traversing the array

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

            // Check if the current element is

            // equal to previous element +1

            if (i > 0 && v.get(i) == v.get(i - 1) + 1)

                count++;

            else

                count = 1;

            // Update the maximum

            ans = Math.max(ans, count);

        }

        return ans;

    }

    // Driver code

    public static void main(String[] args) {

        int arr[] = { 49, 1, 3, 200, 2, 4, 70, 5 };

        int n = arr.length;

        System.out.println(

                "Length of the Longest "

                        + "contiguous subsequence is "

                        + findLongestConseqSubseq(arr, n));

    }

}

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.

package programs;

/\*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\_1 extends Exception {

    public AgeNotWithinRangeException\_1(String message) {

        super(message);

    }

}

class NameNotValidException\_1 extends Exception {

    public NameNotValidException\_1(String message) {

        super(message);

    }

}

class Stud {

    private int rollNo;

    private String name;

    private int age;

    private String course;

    public Stud(int rollNo, String name, int age, String course)

            throws AgeNotWithinRangeException\_1, NameNotValidException\_1 {

        this.rollNo = rollNo;

        if (age < 15 || age > 21) {

            throw new AgeNotWithinRangeException\_1("Age is not within the range of 15 to 21.");

        }

        this.age = age;

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

            throw new NameNotValidException\_1("Name contains invalid characters.");

        }

        this.name = name;

        this.course = course;

    }

    public int getRollNo() {

        return rollNo;

    }

    public String getName() {

        return name;

    }

    public int getAge() {

        return age;

    }

    public String getCourse() {

        return course;

    }

}

public class NameAgeException {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        try {

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

            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();

            Stud student = new Stud(rollNo, name, age, course);

            System.out.println("\nStudent details:");

            System.out.println("Roll No: " + student.getRollNo());

            System.out.println("Name: " + student.getName());

            System.out.println("Age: " + student.getAge());

            System.out.println("Course: " + student.getCourse());

        } catch (AgeNotWithinRangeException\_1 e) {

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

        } catch (NameNotValidException\_1 e) {

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

        }

        scanner.close();

    }

}

25. 

package programs;

//to print all the best in different colors and places

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

class Color\_Demo extends Frame {

     Label lbl1, lbl2, lbl3, lbl4, lbl5;

     public Color\_Demo() {

          lbl1 = new Label("All The Best");

          lbl1.setForeground(Color.red);

          add(lbl1);

          lbl2 = new Label("All The Best");

          lbl2.setForeground(Color.magenta);

          add(lbl2);

          lbl3 = new Label("All The Best");

          lbl3.setForeground(Color.blue);

          add(lbl3);

          lbl4 = new Label("All The Best");

          lbl4.setForeground(Color.green);

          add(lbl4);

          lbl5 = new Label("All The Best");

          lbl5.setForeground(Color.cyan);

          add(lbl5);

          setVisible(true);

          setSize(400, 300);

          setLayout(new FlowLayout());

          setBackground(Color.gray);

     }

     public void paint(Graphics g) {

          g.setColor(Color.magenta);

          g.drawString("All The Best", 100, 100);

          g.setColor(Color.cyan);

          g.drawString("All The Best", 150, 150);

          g.setColor(Color.red);

          g.drawString("All The Best", 200, 200);

          g.setColor(Color.black);

          g.drawString("All The Best", 250, 250);

     }

     public static void main(String[] args) {

          new Color\_Demo();

     }

}