1.

1.1) Program to remove all repeated elements from an array

import java.util.Scanner;

public class Q1\_1RemoveRepeatedInArray {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int size = scanner.nextInt();

int[] array = new int[size];

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

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

array[i] = scanner.nextInt();

}

int[] result = removeRepeatedElements(array);

System.out.println("Array after removing repeated elements:");

for (int element : result) {

System.out.print(element + " ");

}

}

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

int n = array.length;

int[] tempArray = new int[n];

int index = 0;

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

boolean isDuplicate = false;

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

if (array[i] == array[j]) {

isDuplicate = true;

break;

}

}

if (!isDuplicate) {

tempArray[index++] = array[i];

}

}

int[] resultArray = new int[index];

System.arraycopy(tempArray, 0, resultArray, 0, index);

return resultArray;

}

}

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

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class Q1\_2CommonElementsArray {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int size1 = scanner.nextInt();

int[] array1 = new int[size1];

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

for (int i = 0; i < size1; i++) {

array1[i] = scanner.nextInt();

}

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

int size2 = scanner.nextInt();

int[] array2 = new int[size2];

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

for (int i = 0; i < size2; i++) {

array2[i] = scanner.nextInt();

}

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

System.out.println("Common elements between the two arrays:");

for (int element : commonElements) {

System.out.print(element + " ");

}

}

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

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

for (int i = 0; i < array1.length; i++) {

for (int j = 0; j < array2.length; j++) {

if (array1[i] == array2[j] && !commonElementsList.contains(array1[i])) {

commonElementsList.add(array1[i]);

break;

}

}

}

int[] commonElementsArray = new int[commonElementsList.size()];

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

commonElementsArray[i] = commonElementsList.get(i);

}

return commonElementsArray;

}

}

2.

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 Q2\_1CountNoDupString {

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 = Q2\_1CountNoDupString(inputString);

//a Map called wordCountMap is declared,

// which will store words as keys and their corresponding counts as values.

System.out.println("Duplicate words and their counts:");

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

if (entry.getValue() > 1) {

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

}

}

}

public static Map<String, Integer> Q2\_1CountNoDupString(String inputString) {

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

//It starts by splitting the inputString into an array of words using split("\\s+"),

// where "\\s+" is a regular expression matching one or more whitespace characters.

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

//A new HashMap called wordCountMap is created to store the word counts.

for (String word : words) {

wordCountMap.put(word, wordCountMap.getOrDefault(word, 0) + 1);

}

return wordCountMap;

//The overall functionality of this method is to count the occurrences of each word in the input string

// and store the results in a Map where the keys are the words and the values are their respective counts.

}

}

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

import java.util.Scanner;

public class Q2\_2Umbrella {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String inputString = scanner.nextLine();

System.out.print("Enter a character to check: ");

char characterToCheck = scanner.next().charAt(0);

boolean containsCharacter = checkIfContainsCharacter(inputString, characterToCheck);

if (containsCharacter) {

System.out.println("The string contains the character '" + characterToCheck + "'.");

} else {

System.out.println("The string does not contain the character '" + characterToCheck + "'.");

}

}

public static boolean checkIfContainsCharacter(String inputString, char ch) {

for (char c : inputString.toCharArray()) {

if (c == ch) {

return true;

}

}

return false;

}

}

3.

3.1)Java Program to Reverse a String.

import java.util.Scanner;

public class Q3\_1ReverseString {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Prompt the user to enter a string

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

String input = scanner.nextLine();

// Call the reverseString method to reverse the input string

String reversedString = reverseString(input);

// Print the reversed string

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

scanner.close();

}

// Method to reverse the given string

public static String reverseString(String str) {

int length = str.length();

char[] reversedChars = new char[length];

// Iterate over the characters in reverse order and store them in the reversedChars array

for (int i = length - 1, j = 0; i >= 0; i--, j++) {

reversedChars[j] = str.charAt(i);

}

// Create a new string from the reversedChars array

String reversedString = new String(reversedChars);

return reversedString;

}

}

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

import java.util.Scanner;

public class Q3\_2ParlindromeString {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Prompt the user to enter a string

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

String input = scanner.nextLine();

// Check if the input string is a palindrome

boolean isPalindrome = checkPalindrome(input);

// Print the result

if (isPalindrome) {

System.out.println("The string is a palindrome.");

} else {

System.out.println("The string is not a palindrome.");

}

// Close the scanner

scanner.close();

}

// Method to check if the given string is a palindrome

public static boolean checkPalindrome(String str) {

// Remove any spaces and convert the string to lowercase

str = str.replaceAll("\\s", "").toLowerCase();

int left = 0;

int right = str.length() - 1;

// Compare characters from both ends of the string

while (left < right) {

if (str.charAt(left) != str.charAt(right)) {

return false; // Characters don't match, not a palindrome

}

left++;

right--;

}

return true; // All characters matched, it's a palindrome

}

}

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

import java.util.Scanner;

abstract class Vehicle {

abstract void move();

}

class Helicopter extends Vehicle {

@Override

void move() {

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

}

}

class Car extends Vehicle {

@Override

void move() {

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

}

}

class Train extends Vehicle {

@Override

void move() {

System.out.println("The train runs on the track.");

}

}

public class Q4\_Vehicle {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Prompt the user to enter the type of vehicle

System.out.print("Enter the type of vehicle (Helicopter, Car, Train): ");

String vehicleType = scanner.nextLine();

// Create the corresponding vehicle object based on user input

Vehicle vehicle;

switch (vehicleType.toLowerCase()) {

case "helicopter":

vehicle = new Helicopter();

break;

case "car":

vehicle = new Car();

break;

case "train":

vehicle = new Train();

break;

default:

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

scanner.close();

return;

}

// Print the nature of movement of the chosen vehicle

vehicle.move();

// Close the scanner

scanner.close();

}

}

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

import java.util.Scanner;

abstract class Marks {

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

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

double getPercentage() {

double totalMarks = subject1 + subject2 + subject3 + subject4;

return (totalMarks / 400) \* 100;

}

}

public class Q5\_ABStudent\_percentage {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Get marks for Student A

System.out.print("Enter marks for Student A (Subject 1): ");

double marksA1 = scanner.nextDouble();

System.out.print("Enter marks for Student A (Subject 2): ");

double marksA2 = scanner.nextDouble();

System.out.print("Enter marks for Student A (Subject 3): ");

double marksA3 = scanner.nextDouble();

// Create Student A object

A studentA = new A(marksA1, marksA2, marksA3);

// Get marks for Student B

System.out.print("Enter marks for Student B (Subject 1): ");

double marksB1 = scanner.nextDouble();

System.out.print("Enter marks for Student B (Subject 2): ");

double marksB2 = scanner.nextDouble();

System.out.print("Enter marks for Student B (Subject 3): ");

double marksB3 = scanner.nextDouble();

System.out.print("Enter marks for Student B (Subject 4): ");

double marksB4 = scanner.nextDouble();

// Create Student B object

B studentB = new B(marksB1, marksB2, marksB3, marksB4);

// Calculate and print the percentages of marks for both students

double percentageA = studentA.getPercentage();

double percentageB = studentB.getPercentage();

System.out.println("Percentage of Student A: " + percentageA + "%");

System.out.println("Percentage of Student B: " + percentageB + "%");

scanner.close();

}

}

6. Write the following code in your editor below:  
A class named Arithmetic with a method named add that takes 2 integers as parameters and returns an integer denoting their sum.  
A class named Adder that inherits from a superclass named Arithmetic. The main method in the Tester class should print the following: SAMPLE O/P:**My superclass is: Arithmetic  
42 13 20**

class Arithmetic{

public int add(int a, int b){

int sum = a + b;

return sum;

}

}

class Adder extends Arithmetic{

public int callAdd(int a, int b){

return add(a, b);

}

}

class Q6\_Arithmettic\_42\_Tester{

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");

}

}

//import java.util.Scanner;

//

//class Arithmetic {

// public int add(int a, int b) {

// return a + b;

// }

//}

//

//class Adder extends Arithmetic {

// // No additional methods or fields needed in this case

//}

//

//public class Q6\_Arithmettic\_42\_Tester {

// public static void main(String[] args) {

// Scanner sc=new Scanner(System.in);

// Adder adder = new Adder();

//

// // Printing superclass name

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

//

// // Adding two integers

// System.out.println("Enter number 1: ");

// int num1=sc.nextInt();

// System.out.println("Enter number 2: ");

// int num2=sc.nextInt();

// int sum = adder.add(num1, num2);

//

// // Printing the numbers and their sum

// System.out.println(num1 + " " + num2 + " " + sum);

// }

//}

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

import java.util.Scanner;

class MyCalculator {

public long power(int n, int p) throws Exception {

if (n < 0 || p < 0) {

throw new Exception("n or p should not be negative");

} else if (n == 0 && p == 0) {

throw new Exception("n or p should not be zero");

} else {

return (long) Math.pow(n, p);

}

}

}

public class Q7\_ComputePower {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

MyCalculator calculator = new MyCalculator();

System.out.print("Enter the value of n: ");

int n = scanner.nextInt();

System.out.print("Enter the value of p: ");

int p = scanner.nextInt();

try {

long result = calculator.power(n, p);

System.out.println("Result: " + result);

} catch (Exception e) {

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

}

scanner.close();

}

}

8. You are given a phone book that consists of people's names and their phone number. After that you will be given some person's name as query. For each query, print the phone number of that person. Use HashMap to implement it.The first line will have an integer denoting the number of entries in the phone book. Each entry consists of two lines: a name and the corresponding phone number.  
After these, there will be some queries. Each query will contain a person's name. Read the queries until end-of-file.  
Constraints:  
A person's name consists of only lower-case English letters and it may be in the format 'first-name last-name' or in the format 'first-name'. Each phone number has exactly 8 digits without any leading zeros.For each case, print "Not found" if the person has no entry in the phone book. Otherwise, print the person's name and phone number.

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class Q8\_Phonebook {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Prompt for the number of phone book entries

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

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<>();

// Enter 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.toLowerCase(), phoneNumber);

}

//Process the queries

System.out.println("Enter the queries (Enter 'exit' to quit):");

String query;

while (true) {

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

query = scanner.nextLine();

if (query.equalsIgnoreCase("exit")) {

break;

}

String phoneNumber = phoneBook.get(query.toLowerCase());

if (phoneNumber != null) {

System.out.println("Name: " + query + ", Phone Number: " + phoneNumber);

} else {

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

}

}

scanner.close();

}

}

// // Process a single query

// System.out.print("Enter the name to search: ");

// String query = scanner.nextLine();

//

// String phoneNumber = phoneBook.get(query.toLowerCase());

// if (phoneNumber != null) {

// System.out.println("Name: " + query + ", Phone Number: " + phoneNumber);

// } else {

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

// }

//

// scanner.close();

// }

//}

//import java.util.HashMap;

//import java.util.Map;

//import java.util.Scanner;

//

//public class Q8\_Phonebook {

// public static void main(String[] args) {

// Scanner scanner = new Scanner(System.in);

//

// // Create a HashMap to store the phone book entries

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

//

// // Add some example phone book entries

// phoneBook.put("siddhi", "12345678");

// phoneBook.put("manasi", "87654321");

// phoneBook.put("geeta", "45678912");

// phoneBook.put("karishma kapoor", "98765432");

// phoneBook.put("riya patil", "23456789");

//

// // Prompt for the name to search

// System.out.print("Enter the name to search: ");

// String query = scanner.nextLine().toLowerCase();

//

// String phoneNumber = phoneBook.get(query);

// if (phoneNumber != null) {

// System.out.println("Name: " + query + ", Phone Number: " + phoneNumber);

// } else {

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

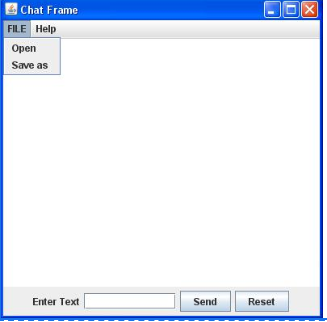
// }

//

// scanner.close();

// }

//}

9. 

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Q9\_ChatFrame {

public static void main(String[] args) {

JFrame jf = new JFrame();

jf.setSize(500, 500);

JPanel panel = new JPanel();

JMenuBar menuBar = new JMenuBar();

JMenu file = new JMenu("File");

JMenu help = new JMenu("Help");

menuBar.add(file);

menuBar.add(help);

JMenuItem open = new JMenuItem("Open");

JMenuItem saveAs = new JMenuItem("Save as");

file.add(open);

file.add(saveAs);

JTextArea textArea = new JTextArea();

JTextField text = new JTextField(15);

JButton reset = new JButton("Reset");

JButton send = new JButton("Send");

JLabel label = new JLabel("Please Enter Text");

panel.add(label);

panel.add(text);

panel.add(send);

panel.add(reset);

// ActionListener for the Send button

send.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String enteredText = text.getText();

textArea.append(enteredText + "\n");

text.setText("");

}

});

// ActionListener for the Reset button

reset.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

textArea.setText("");

}

});

jf.getContentPane().add(menuBar, BorderLayout.NORTH);

jf.getContentPane().add(panel, BorderLayout.SOUTH);

jf.getContentPane().add(textArea, BorderLayout.CENTER);

jf.setVisible(true);

}

}

10. 

//import javax.swing.\*;

//import java.awt.\*;

//

//public class Q10\_NumberAddition {

// public static void main(String[] args) {

// JFrame f = new JFrame("Number Addition");

//

// JLabel l = new JLabel("Number Addition");

// l.setBounds(30,50,100,20);

// l.setBackground(Color.BLUE);

//

// JLabel l1 = new JLabel("First Number:");

// l1.setBounds(45,80,90,20);

//

// JTextField t = new JTextField();

// t.setBounds(145,80,100,20);

// t.setEditable(true);

//

// JLabel l2 = new JLabel("Second Number:");

// l2.setBounds(45,120,125,20);

//

// JTextField t1 = new JTextField();

// t1.setBounds(145,120,100,20);

// t1.setEditable(true);

//

// JLabel l3 = new JLabel("Result:");

// l3.setBounds(75,160,125,20);

//

// JTextField t2 = new JTextField();

// t2.setBounds(125,160,100,20);

// t2.setEditable(true);

//

// JButton b = new JButton("Add");

// b.setBounds(125,225,90,20);

//

// JButton b1 = new JButton("Clear");

// b1.setBounds(225,225,90,20);

//

// JButton b2 = new JButton("Exit");

// b2.setBounds(30,300,90,20);

//

//

//

//

// f.add(l);

// f.add(l1);

// f.add(t);

// f.add(l2);

// f.add(t1);

// f.add(l3);

// f.add(t2);

// f.add(b);

// f.add(b1);

// f.add(b2);

// f.setSize(400,400);

// f.setLayout(null);

// f.setVisible(true);

// }

//}

//import java.awt.EventQueue;

//import javax.swing.\*;

//import java.awt.\*;

//public class Q10\_NumberAddition {

// private JFrame frame;

// /\*\*

// \* Launch the application.

// \*/

// public static void main(String[] args) {

// EventQueue.invokeLater(new Runnable() {

// public void run() {

// try {

// Q10\_NumberAddition window = new Q10\_NumberAddition();

// window.frame.setVisible(true);

// } catch (Exception e) {

// e.printStackTrace();

// }

// }

// });

// }

// /\*\*

// \* Create the application.

// \*/

// public Q10\_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);

// //for number 1

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

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

// frame.getContentPane().add(first);

// JTextField f1=new JTextField();

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

// frame.getContentPane().add(f1);

// //for number 2

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

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

// frame.getContentPane().add(second);

// JTextField s1=new JTextField();

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

// frame.getContentPane().add(s1);

// //for result

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

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

// frame.getContentPane().add(result);

// JTextField r1=new JTextField();

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

// frame.getContentPane().add(r1);

// //add button

// JButton add=new JButton("ADD");

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

// frame.getContentPane().add(add);

// //clear button

// JButton clear=new JButton("CLEAR");

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

// frame.getContentPane().add(clear);

// //exit button

// JButton exit=new JButton("Exit");

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

// frame.add(exit);

// }

//}

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Q10\_NumberAddition extends JFrame implements ActionListener {

public static final int WIDTH = 400;

public static final int HEIGHT = 400;

private JTextField firstNumberField;

private JTextField secondNumberField;

private JTextField resultField;

private double sum = 0;

public static void main(String[] args) {

Q10\_NumberAddition guiAdder = new Q10\_NumberAddition();

guiAdder.setVisible(true);

}

public Q10\_NumberAddition() {

setTitle("Adding Machine");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); // Exit the application when the window is closed

setSize(WIDTH, HEIGHT);

Container contentPane = getContentPane();

contentPane.setLayout(new BorderLayout());

JPanel inputPanel = new JPanel();

inputPanel.setLayout(new GridLayout(3, 2));

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

firstNumberField = new JTextField(10);

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

secondNumberField = new JTextField(10);

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

resultField = new JTextField(10);

resultField.setEditable(false);

inputPanel.add(firstNumberLabel);

inputPanel.add(firstNumberField);

inputPanel.add(secondNumberLabel);

inputPanel.add(secondNumberField);

inputPanel.add(resultLabel);

inputPanel.add(resultField);

contentPane.add(inputPanel, BorderLayout.NORTH);

JPanel buttonPanel = new JPanel();

buttonPanel.setBackground(Color.WHITE);

buttonPanel.setLayout(new FlowLayout(FlowLayout.RIGHT));

JButton addButton = new JButton("Add");

addButton.addActionListener(this);

JButton resetButton = new JButton("Clear");

resetButton.addActionListener(this);

buttonPanel.add(addButton);

buttonPanel.add(resetButton);

contentPane.add(buttonPanel, BorderLayout.CENTER);

JButton exitButton = new JButton("Exit");

exitButton.addActionListener(this);

JPanel exitPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));

exitPanel.add(exitButton);

contentPane.add(exitPanel, BorderLayout.SOUTH);

}

public void actionPerformed(ActionEvent e) {

if (e.getActionCommand().equals("Add")) {

try {

double firstNumber = Double.parseDouble(firstNumberField.getText());

double secondNumber = Double.parseDouble(secondNumberField.getText());

sum = firstNumber + secondNumber;

resultField.setText(Double.toString(sum));

} catch (NumberFormatException ex) {

resultField.setText("Invalid input");

}

} else if (e.getActionCommand().equals("Reset")) {

firstNumberField.setText("");

secondNumberField.setText("");

resultField.setText("");

} else if (e.getActionCommand().equals("Exit")) {

System.exit(0);

} else {

resultField.setText("Error in adder code.");

}

}

}

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 Q11\_1MultiplicationTable {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Prompt the user to enter a number

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

int number = scanner.nextInt();

// Print the multiplication table

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

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

int result = number \* i;

System.out.println(number + " x " + i + " = " + result);

}

scanner.close();

}

}

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

import java.util.Scanner;

public class Q11\_2PrimeCheck {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

if (n<=1) {

System.out.println("The number is neither prime nor composite.");

} else if (checkPrime(n)) {

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

} else {

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

}

scanner.close();

}

public static boolean checkPrime(int n) {

int m = n / 2;

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

if (n % i == 0) {

return false;

}

}

return true;

}

}

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 Q12\_DiamondPattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Input number of rows (half): ");

int rows = scanner.nextInt();

// Upper half of the pattern

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

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

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

}

System.out.println();

}

// Lower half of the pattern

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

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

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

}

System.out.println();

}

scanner.close();

}

}

//full diamond

//import java.util.Scanner;

//public class Q12\_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();

//

// // 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();

// }

//

// scanner.close();

// }

//}

13.

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

import java.util.Scanner;

public class Q13\_TransposeMatrix {

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: ");

int rows = scanner.nextInt();

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

int columns = scanner.nextInt();

// Create the matrix

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

// Read the elements of the matrix

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();

}

}

// Find the transpose of 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.util.Scanner;

public class Q13\_2WordsInTextFile {

public static void main(String[] args) {

try {

File file = new File("C:\\Users\\Desktop\\try.txt"); // Replace with the actual path to your text file

Scanner scanner = new Scanner(file);

int wordCount = 0;

while (scanner.hasNext()) {

String word = scanner.next();

wordCount++;

}

scanner.close();

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

} catch (FileNotFoundException e) {

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

e.printStackTrace();

}

}

}

14. 

import java.awt.event.\*;

import java.awt.\*;

import javax.swing.\*;

public class Q14\_SwingCalculator extends JFrame implements ActionListener

{

JButton b10,b11,b12,b13,b14,b15;

JButton b[]=new JButton[10];

int i,r,n1,n2;

JTextField res;

char op;

public Q14\_SwingCalculator()

{

super("Calulator");

setLayout(new BorderLayout());

JPanel p=new JPanel();

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

for(int i=0;i<=9;i++)

{

b[i]=new JButton(i+"");

p.add(b[i]);

b[i].addActionListener(this);

}

b10=new JButton("+");

p.add(b10);

b10.addActionListener(this);

b11=new JButton("-");

p.add(b11);

b11.addActionListener(this);

b12=new JButton("\*");

p.add(b12);

b12.addActionListener(this);

b13=new JButton("/");

p.add(b13);

b13.addActionListener(this);

b14=new JButton("=");

p.add(b14);

b14.addActionListener(this);

b15=new JButton("C");

p.add(b15);

b15.addActionListener(this);

res=new JTextField(10);

add(p,BorderLayout.CENTER);

add(res,BorderLayout.NORTH);

setVisible(true);

setSize(200,200);

}

public void actionPerformed(ActionEvent ae)

{

JButton pb=(JButton)ae.getSource();

if(pb==b15)

{

r=n1=n2=0;

res.setText("");

}

else

if(pb==b14)

{

n2=Integer.parseInt(res.getText());

eval();

res.setText(""+r);

}

else

{

boolean opf=false;

if(pb==b10)

{ op='+';

opf=true;

}

if(pb==b11)

{ op='-';opf=true;}

if(pb==b12)

{ op='\*';opf=true;}

if(pb==b13)

{ op='/';opf=true;}

if(opf==false)

{

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

{

if(pb==b[i])

{

String t=res.getText();

t+=i;

res.setText(t);

}

}

}

else

{

n1=Integer.parseInt(res.getText());

res.setText("");

}

}

}

int eval()

{

switch(op)

{

case '+': r=n1+n2; break;

case '-': r=n1-n2; break;

case '\*': r=n1\*n2; break;

case '/': r=n1/n2; break;

}

return 0;

}

public static void main(String arg[])

{

new Q14\_SwingCalculator();

}

}

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 Q15\_IterateArrayList {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

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

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

System.out.println("Enter the element " + (i + 1) + ": ");

int number = scanner.nextInt();

numbers.add(number);

}

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

iterateUsingIterator(numbers);

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

iterateUsingAdvancedForLoop(numbers);

System.out.println("\nFor Loop:");

iterateUsingForLoop(numbers);

}

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

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

while (iterator.hasNext()) {

int number = iterator.next();

System.out.println(number);

}

}

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

for (int number : list) {

System.out.println(number);

}

}

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

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

int number = list.get(i);

System.out.println(number);

}

}

}

////taking input hardcode

//import java.util.ArrayList;

//import java.util.Iterator;

//

//public class Q15\_IterateArrayList {

// public static void main(String[] args) {

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

//

// numbers.add(20);

// numbers.add(30);

// numbers.add(40);

//

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

// iterateUsingIterator(numbers);

//

// System.out.println("\nAdvanced For Loop:");

// iterateUsingAdvancedForLoop(numbers);

//

// System.out.println("\nFor Loop:");

// iterateUsingForLoop(numbers);

// }

//

// public static void iterateUsingIterator(ArrayList<Integer> list) {

// Iterator<Integer> iterator = list.iterator();

//

// while (iterator.hasNext()) {

// int number = iterator.next();

// System.out.println(number);

// }

// }

//

// public static void iterateUsingAdvancedForLoop(ArrayList<Integer> list) {

// for (int number : list) {

// System.out.println(number);

// }

// }

//

// public static void iterateUsingForLoop(ArrayList<Integer> list) {

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

// int number = list.get(i);

// System.out.println(number);

// }

// }

//}

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 Q16\_WordCountHashMap {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String input = scanner.nextLine();

scanner.close();

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

System.out.println(wordCountMap);

}

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

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

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

for (String word : words) {

wordCountMap.put(word, wordCountMap.getOrDefault(word, 0) + 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 .

import java.util.Arrays;

import java.util.Scanner;

public class Q17\_10StringsFromConsole {

public static void main(String[] args) {

// Read 10 strings from console

Scanner scanner = new Scanner(System.in);

String[] strings = new String[10];

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

System.out.print("Enter string " + (i+1) + ": ");

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

String combinedString = strings[0] + strings[1];

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

// Reverse the first string

String reversedString = new StringBuilder(strings[0]).reverse().toString();

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

}

}

////not direct sorting

// // Sort the strings

// sortStrings(strings);

//

// // Print the sorted strings

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

// for (String str : strings) {

// System.out.println(str);

// }

// }

// // Helper method to sort an array of strings

// private static void sortStrings(String[] strings) {

// for (int i = 0; i < strings.length - 1; i++) {

// for (int j = i + 1; j < strings.length; j++) {

// if (strings[i].compareTo(strings[j]) > 0) {

// String temp = strings[i];

// strings[i] = strings[j];

// strings[j] = temp;

// }

// }

// }

// }

//}

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 setData(String name, int age) {

this.name = name;

this.age = age;

}

}

class Employee extends Person {

private String designation;

private double salary;

public void setData(String name, int age, String designation, double salary) {

super.setData(name, age);

this.designation = designation;

this.salary = salary;

}

public double getSalary() {

return salary;

}

public String getName() {

return name;

}

}

public class Q18\_Salary5Persons {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int numOfEmployees = scanner.nextInt();

scanner.nextLine();

Employee[] employees = new Employee[numOfEmployees];

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

System.out.println("Enter details for Person " + (i + 1) + ":");

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

String name = scanner.nextLine();

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

int age = scanner.nextInt();

scanner.nextLine();

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

String designation = scanner.nextLine();

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

double salary = scanner.nextDouble();

scanner.nextLine();

employees[i] = new Employee();

employees[i].setData(name, age, designation, salary);

}

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

for (Employee employee : employees) {

if (employee.getSalary() > 5000) {

System.out.println(employee.getName());

}

}

scanner.close();

}

}

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 Person {

void store(String name);

void disp();

}

// Customer class implementing Account and Person interfaces

class Customer implements Account, Person {

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

class Q19\_MultiInheritance {

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.  
"



class VideoTape {

private String title;

private int duration;

public VideoTape(String title, int duration) {

this.title = title;

this.duration = duration;

}

public String getTitle() {

return title;

}

public int getDuration() {

return duration;

}

public void display() {

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

System.out.println("Duration: " + duration + " minutes");

}

}

class Movie extends VideoTape {

private String director;

private String genre;

public Movie(String title, int duration, String director, String genre) {

super(title, duration);

this.director = director;

this.genre = genre;

}

public String getDirector() {

return director;

}

public String getGenre() {

return genre;

}

@Override

public void display() {

super.display();

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

System.out.println("Genre: " + genre);

}

}

class MusicVideo extends VideoTape {

private String artist;

private String album;

public MusicVideo(String title, int duration, String artist, String album) {

super(title, duration);

this.artist = artist;

this.album = album;

}

public String getArtist() {

return artist;

}

public String getAlbum() {

return album;

}

@Override

public void display() {

super.display();

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

System.out.println("Album: " + album);

}

}

public class Q20\_Hierarchy {

public static void main(String[] args) {

Movie movie = new Movie("Inception", 148, "Christopher Nolan", "Science Fiction");

movie.display();

System.out.println();

MusicVideo musicVideo = new MusicVideo("Shape of You", 233, "Ed Sheeran", "Divide");

musicVideo.display();

}

}

class VideoTape {

protected String title;

protected int duration;

public VideoTape(String title, int duration) {

this.title = title;

this.duration = duration;

}

public void play() {

System.out.println("Playing video tape: " + title);

}

public void stop() {

System.out.println("Stopping video tape: " + title);

}

}

class Movie extends VideoTape {

private String director;

private int year;

public Movie(String title, int duration, String director, int year) {

super(title, duration);

this.director = director;

this.year = year;

}

public void displayInfo() {

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

System.out.println("Duration: " + duration + " minutes");

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

System.out.println("Year: " + year);

}

}

class MusicVideo extends VideoTape {

private String artist;

private String album;

public MusicVideo(String title, int duration, String artist, String album) {

super(title, duration);

this.artist = artist;

this.album = album;

}

public void displayInfo() {

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

System.out.println("Duration: " + duration + " minutes");

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

System.out.println("Album: " + album);

}

}

public class Q20\_Hierarchy {

public static void main(String[] args) {

Movie movie = new Movie("The Matrix", 136, "Lana Wachowski", 1999);

movie.displayInfo();

movie.play();

movie.stop();

System.out.println();

MusicVideo musicVideo = new MusicVideo("Bohemian Rhapsody", 354, "Queen", "A Night at the Opera");

musicVideo.displayInfo();

musicVideo.play();

musicVideo.stop();

}

}

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.Scanner;

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) {

String courseName = course.toLowerCase(); // Convert course name to lowercase

courses.add(courseName);

}

public void removeCourse(String course) {

String courseName = course.toLowerCase(); // Convert course name to lowercase

courses.remove(courseName);

}

public void displayInfo() {

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

System.out.println("Grade: " + grade);

System.out.println("Courses: " + courses);

}

}

public class Q21\_StudentCourses {

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();

Student student = new Student(name, grade);

System.out.println("Enter courses (enter 'done' to finish):");

String course;

while (true) {

course = scanner.nextLine();

if (course.equalsIgnoreCase("done")) { // Case-insensitive check for 'done'

break;

}

student.addCourse(course);

}

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

student.displayInfo();

System.out.print("\nEnter a course to remove: ");

String courseToRemove = scanner.nextLine();

student.removeCourse(courseToRemove);

System.out.println("\nUpdated Student Information:");

student.displayInfo();

while (true) {

System.out.print("\nDo you want to add more courses? (yes/no): ");

String choice = scanner.nextLine();

if (choice.equalsIgnoreCase("yes")) {

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

course = scanner.nextLine();

student.addCourse(course);

System.out.println("\nUpdated Student Information:");

student.displayInfo();

} else {

break;

}

}

scanner.close();

}

}

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 Person {

protected String firstName;

protected String lastName;

public Person(String firstName, String lastName) {

this.firstName = firstName;

this.lastName = lastName;

}

public String getFirstName() {

return firstName;

}

public String getLastName() {

return lastName;

}

}

class Employee extends Person {

private String employeeId;

private String jobTitle;

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

super(firstName, lastName);

this.employeeId = employeeId;

this.jobTitle = jobTitle;

}

public String getEmployeeId() {

return employeeId;

}

@Override

public String getLastName() {

return lastName + " (" + jobTitle + ")";

}

}

public class Q22\_PersonEmployeeJob {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String firstName = scanner.nextLine();

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

String lastName = scanner.nextLine();

System.out.print("Enter employee ID: ");

String employeeId = scanner.nextLine();

System.out.print("Enter employee's job title: ");

String jobTitle = scanner.nextLine();

Employee employee = new Employee(firstName, lastName, employeeId, jobTitle);

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

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

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

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

scanner.close();

}

}

23. Write a Java program to find the length of the longest consecutive elements sequence from an unsorted array of integers.  
Sample array: [49, 1, 3, 200, 2, 4, 70, 5]  
The longest consecutive elements sequence is [1, 2, 3, 4, 5], therefore the program will return its length 5.

import java.util.HashSet;

import java.util.Scanner;

public class Q23\_LongestConsecutiveElements {

public static int[] findLongestConsecutiveSequence(int[] nums) {

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

for (int num : nums) {

set.add(num);

}

int maxLength = 0;

int currentLength = 0;

int startNum = 0;

for (int num : nums) {

if (!set.contains(num - 1)) {

int currentNum = num;

currentLength = 1;

while (set.contains(currentNum + 1)) {

currentNum++;

currentLength++;

}

if (currentLength > maxLength) {

maxLength = currentLength;

startNum = num;

}

}

}

int[] sequence = new int[maxLength];

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

sequence[i] = startNum + i;

}

return sequence;

}

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[] nums = new int[n];

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

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

nums[i] = scanner.nextInt();

}

int[] longestSequence = findLongestConsecutiveSequence(nums);

int sequenceLength = longestSequence.length;

// System.out.println("The longest consecutive elements sequence is:");

// for (int i = 0; i < sequenceLength; i++) {

// System.out.print(longestSequence[i]);

// if (i != sequenceLength - 1) {

// System.out.print(", ");

// }

// }

System.out.println("\nLength of the longest consecutive sequence is: " + sequenceLength);

scanner.close();

}

}

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

import java.util.Scanner;

class AgeNotWithinRangeException extends Exception {

public AgeNotWithinRangeException(String message) {

super(message);

}

}

class NameNotValidException extends Exception {

public NameNotValidException(String message) {

super(message);

}

}

class Student {

private int rollNo;

private String name;

private int age;

private String course;

public Student(int rollNo, String name, int age, String course) throws AgeNotWithinRangeException, NameNotValidException {

if (age < 15 || age > 21) {

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

}

if (name.matches(".\*\\d.\*") || !name.matches("[a-zA-Z\\s]+")) {

throw new NameNotValidException("Name contains numbers or special symbols");

}

this.rollNo = rollNo;

this.name = name;

this.age = age;

this.course = course;

}

public void displayInfo() {

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

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

System.out.println("Age: " + age);

System.out.println("Course: " + course);

}

}

public class Q24\_StudentNameExceptions {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

try {

System.out.print("Enter Roll No: ");

int rollNo = scanner.nextInt();

scanner.nextLine();

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

String name = scanner.nextLine();

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

int age = scanner.nextInt();

scanner.nextLine();

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

String course = scanner.nextLine();

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

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

student.displayInfo();

} catch (AgeNotWithinRangeException e) {

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

} catch (NameNotValidException e) {

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

} catch (Exception e) {

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

}

scanner.close();

}

}

25. 

import java.awt.\*;

class Q25\_AllTheBest extends Frame

{

Label lbl1,lbl2,lbl3,lbl4,lbl5;

public Q25\_AllTheBest()

{

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 Q25\_AllTheBest();

}

}

LAB 1

public class Participant{

String name;

String registrationId;

long contactNumber;

String branch;

static int count;

Participant(String name,long contactNumber,String branch){

this.name=name;

this.contactNumber=contactNumber;

this.branch=branch;

Participant.count++;

registrationId="D"+String.valueOf(count);

}

static{

count=1000;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public long getContactNumber() {

return contactNumber;

}

public void setContactNumber(long contactNumber) {

this.contactNumber = contactNumber;

}

public String getBranch() {

return branch;

}

public void setBranch(String branch) {

this.branch = branch;

}

public static int getCount() {

return count;

}

public static void setCount(int count) {

Participant.count = count;

}

public String getRegistrationId() {

return registrationId;

}

public static void main(String[] args){

Participant p1=new Participant("Rohit",1234567899,"Computer");

Participant p2=new Participant("Sayli",1988612300,"Mechanical");

System.out.println("Hi "+ p1.name+"! Your registration id is "+ p1.registrationId + "\n");

System.out.println("Hi "+ p2.name+"! Your registration id is "+ p2.registrationId);

}

}

LAB 2

import java.util.Scanner;

public class Lab2 {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int length;

int l=0;

int element;

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

length=sc.nextInt();

int arr[]=new int[length];

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

l=0;

System.out.printf("Enter element %d: ",i+1);

element = sc.nextInt();

for(int j:arr)

{

if(j==element){

l++;

}

}

if(l==0){

arr[i]=element;

}

else{

System.out.println("Element already exists, enter a unique element: ");

i--;

}

}

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

for (int p=0;p<length;p++){

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

}

System.out.println("\nEntered the desired sum: ");

int numsum=sc.nextInt();

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

int x=arr[i];

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

int y=arr[j];

if((x+y==numsum) && (x!=y)){

System.out.println("Pair exists: ["+x+", "+y+"]");

}

}

}

}

}

LAB 3

import java.util.Scanner;

class Shape

{

double area;

void setarea(double r){

area=3.14\*r\*r;

System.out.println("The area of circle is:"+area);

}

void setarea(int side){

area=side\*side;

System.out.println("The area of square is:"+area);

}

void setarea(int b,int h){

area=0.5\*b\*h;

System.out.println("The area of triangle is:"+area);

}

}

class Circle extends Shape

{

void getInput(){

double r;

System.out.println("Enter the radius of the circle : ");

Scanner sc = new Scanner(System.in);

r=sc.nextInt();

setarea(r);

}

}

class Triangle extends Shape

{

void getInput(){

int b;

int h;

System.out.println("Enter the base and height of Triangle : ");

Scanner sc=new Scanner (System.in);

b=sc.nextInt();

h=sc.nextInt();

setarea(b,h);

}

}

class Square extends Shape

{

void getInput(){

int side;

System.out.println("Enter the side of square: ");

Scanner sc=new Scanner (System.in);

side=sc.nextInt();

setarea(side);

}

}

class Tester{

public static void main(String[] args) {

Circle c=new Circle();

c.getInput();

Square s=new Square() ;

s.getInput();

Triangle t=new Triangle();

t.getInput();

}

}

LAB 4

import java.util.Scanner;

public class Lab4{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

try{

System.out.println("Input the numerator: ");

int a=sc.nextInt();

System.out.println("Input the numerator: ");

int b=sc.nextInt();

int c=a/b;

System.out.println(c);

}

catch(ArithmeticException e){

System.out.println(e);

System.out.println("You should not divide a number by zero.");

}

try{

System.out.println("Enter Array Size: ");

int size=sc.nextInt();

int[] arr= new int[size];

System.out.println("Enter the index: ");

int ind=sc.nextInt();

arr[ind]=5;

}

catch(ArrayIndexOutOfBoundsException e){

System.out.println(e);

System.out.println("OOPs!!!Array Index out of bounds for the given size");

}

try{

String sr=null;

int len=sr.length();

}

catch(NullPointerException e){

System.out.println(e);

System.out.println("Null Pointer Exception arises!!");

}

}

}

LAB 5

import java.util.ArrayList;

class Order {

private int orderId;

private ArrayList<String> itemNames;

private boolean isDelivered;

public Order(int orderId, ArrayList<String> itemNames, boolean isDelivered) {

this.orderId = orderId;

this.itemNames = itemNames;

this.isDelivered = isDelivered;

}

public ArrayList<String> getItemNames() {

return itemNames;

}

public boolean isDelivered() {

return isDelivered;

}

}

class Tester {

public static void main(String[] args) {

ArrayList<Order> orders = new ArrayList<Order>();

orders.add(new Order(101, new ArrayList<String>() {{

add("Jeans");

add("Shirt");

add("Belt");

}}, true));

orders.add(new Order(102, new ArrayList<String>() {{

add("Tie");

add("Shirt");

}}, true));

orders.add(new Order(103, new ArrayList<String>() {{

add("Tshirt");

add("Socks");

add("Tie");

}}, true));

ArrayList<String> items = getItems(orders);

for (String item : items) {

System.out.print(item + ",");

}

}

public static ArrayList<String> getItems(ArrayList<Order> orders) {

ArrayList<String> items = new ArrayList<String>();

for (Order order : orders) {

if (order.isDelivered()) {

items.addAll(order.getItemNames());

} //if ordered delivered then return items

}

return items;

}

}

LAB 6

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.\*;

class Lab6 {

public static void main(String[] args) {

JFrame f=new JFrame("Vishwakarma Institue of Technology");

JLabel l1=new JLabel("Name");

l1.setBounds(100,100,80,30);

JLabel l2=new JLabel("Roll number");

l2.setBounds(100,150,80,30);

JLabel l3=new JLabel("Branch");

l3.setBounds(100,200,80,30);

JLabel l4=new JLabel("PRN number");

l4.setBounds(90,250,200,30);

JLabel l5=new JLabel("Year");

l5.setBounds(100,300,80,30);

JTextField t1=new JTextField();

t1.setBounds(200,100,100,30);

JTextField t2=new JTextField();

t2.setBounds(200,150,100,30);

JTextField t3=new JTextField();

t3.setBounds(200,250,100,30);

JTextField t4=new JTextField();

t4.setBounds(200,300,100,30);

String branch[]={"Computer","IT","Mechanical","Chemical","Production","ENTC","AIDS"};

JComboBox c1=new JComboBox<>(branch);

c1.setBounds(200,200,100,30);

JButton b = new JButton("Submit");

b.setBounds(200,400,100,30);

JPanel p=new JPanel();

p.setBounds(200,300,100,30);

f.add(t4);

f.add(l1);

f.add(l2);

f.add(l3);

f.add(l4);

f.add(l5);

f.add(t1);

f.add(t2);

f.add(t3);

f.add(p);

f.add(c1);

f.add(b);

f.setLayout(null);

f.setSize(500,500);

f.setVisible(true);

b.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e1) {

try

{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection c= DriverManager.getConnection("jdbc:mysql://localhost:3306/database or tablename","root","Your MySQL password");

Statement st = c.createStatement();

String name=t1.getText();

String rollno=t2.getText();

String branch=(String) c1.getSelectedItem();

String prn=t3.getText();

String year=t4.getText();

String querry=String.format("INSERT INTO registration\_form VALUES('%s','%s','%s','%s','%s')",name,rollno,branch,prn,year);

st.executeUpdate(querry);

System.out.println("Querry Executed");

} catch (ClassNotFoundException | SQLException e) {

throw new RuntimeException(e);

}

}

});

}

}

Maatrix Addition

import java.util.Scanner;

public class MatrixAddition {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int rows = scanner.nextInt();

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

int columns = scanner.nextInt();

int[][] matrix1 = new int[rows][columns];

int[][] matrix2 = new int[rows][columns];

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

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

enterMatrixElements(scanner, matrix1);

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

enterMatrixElements(scanner, matrix2);

addMatrices(matrix1, matrix2, sumMatrix);

System.out.println("Sum of the matrices:");

displayMatrix(sumMatrix);

}

public static void enterMatrixElements(Scanner scanner, int[][] matrix) {

int rows = matrix.length;

int columns = matrix[0].length;

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

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

System.out.print("Enter element at position (" + (i + 1) + "," + (j + 1) + "): ");

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

}

}

}

public static void addMatrices(int[][] matrix1, int[][] matrix2, int[][] sumMatrix) {

int rows = matrix1.length;

int columns = matrix1[0].length;

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

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

sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];

}

}

}

public static void displayMatrix(int[][] matrix) {

int rows = matrix.length;

int columns = matrix[0].length;

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

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

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

}

System.out.println();

}

}

}