1.

1.1) Program to remove all repeated elements from an array

import java.util.Scanner;

public class RemoveDuplicateInArray {

public static int removeDuplicateElements(int arr[], int n) {

if (n == 0 || n == 1) {

return n;

}

int[] temp = new int[n];

int j = 0;

for (int i = 0; i < n - 1; i++) {

if (arr[i] != arr[i + 1]) {

temp[j++] = arr[i];

}

}

temp[j++] = arr[n - 1];

// Changing original array

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

arr[i] = temp[i];

}

return j;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int length = scanner.nextInt();

int arr[] = new int[length];

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

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

arr[i] = scanner.nextInt();

}

length = removeDuplicateElements(arr, length);

// Printing array elements

System.out.println("Array with duplicates removed:");

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

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

}

}

}

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

import java.util.Arrays;

import java.util.Scanner;

public class CommonElements {

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 length1 = scanner.nextInt();

int[] array1 = new int[length1];

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

for (int i = 0; i < length1; i++) {

array1[i] = scanner.nextInt();

}

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

int length2 = scanner.nextInt();

int[] array2 = new int[length2];

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

for (int i = 0; i < length2; i++) {

array2[i] = scanner.nextInt();

}

System.out.println("Array1: " + Arrays.toString(array1));

System.out.println("Array2: " + Arrays.toString(array2));

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

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

if (array1[i] == array2[j]) {

System.out.println("Common element is: " + array1[i]);

}

}

}

}

}

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

import java.util.Scanner;

public class DuplicateWord {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String string = scanner.nextLine();

int count;

// Converts the string into lowercase

string = string.toLowerCase();

// Split the string into words using built-in function

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

System.out.println("Duplicate words in the given string:");

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

count = 1;

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

if (words[i].equals(words[j])) {

count++;

// Set words[j] to "0" to avoid printing the visited word

words[j] = "0";

}

}

// Displays the duplicate word if count is greater than 1

if (count > 1 && !words[i].equals("0")) {

System.out.println(words[i]);

}

}

}

}

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

import java.util.Scanner;

public class CheckLetter {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String str = scanner.nextLine();

char letter;

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

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

boolean present = false;

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

if (str.charAt(i) == letter) {

present = true;

break;

}

}

System.out.println("The letter '" + letter + "' is present in the string: " + present);

}

}

3.

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 input = scanner.nextLine();

String reversed = reverseString(input);

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

}

public static String reverseString(String str) {

StringBuilder sb = new StringBuilder(str);

return sb.reverse().toString();

}

}

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

import java.util.Scanner;

class palindrome {

public static void main(String[] args) {

Scanner scanner=new Scanner(System.in);

System.out.println("Enter a string as an input to check whether it is palindrome or not");

String input= scanner.nextLine();

//checking whether palindrome or not

if(isPalindrome(input))

{

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

}

else

{

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

}

}

public static boolean isPalindrome(String str) {

int left = 0, right = str.length() - 1;

while(left < right)

{

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

import java.util.Scanner;

interface Vehicle {

void move();

}

class Helicopter implements Vehicle {

//@Override

public void move() {

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

}

}

class Car implements Vehicle {

//@Override

public void move() {

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

}

}

class Train implements Vehicle {

//@Override

public void move() {

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

}

}

public class VehicleFactory {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String vehicleType = scanner.nextLine().toLowerCase();

Vehicle vehicle = createVehicle(vehicleType);

if (vehicle != null) {

vehicle.move();

} else {

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

}

}

public static Vehicle createVehicle(String vehicleType) {

switch (vehicleType) {

case "helicopter":

return new Helicopter();

case "car":

return new Car();

case "train":

return new Train();

default:

return null;

}

}

}

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

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 Percentage {

public static void main(String[] args) {

A studentA = new A(85, 90, 92);

B studentB = new B(75, 80, 85, 88);

double percentageA = studentA.getPercentage();

double percentageB = studentB.getPercentage();

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

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

}

}

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.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 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 and p should not be zero");

} else {

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

}

}

}

public class power {

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.

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class phonebook {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int n = scanner.nextInt();

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

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

// Read the phone book entries

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

String name = scanner.nextLine();

String phoneNumber = scanner.nextLine();

phoneBook.put(name, phoneNumber);

}

// Process the queries

while (scanner.hasNextLine()) {

String query = scanner.nextLine();

if (phoneBook.containsKey(query)) {

System.out.println(query + " " + phoneBook.get(query));

} else {

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

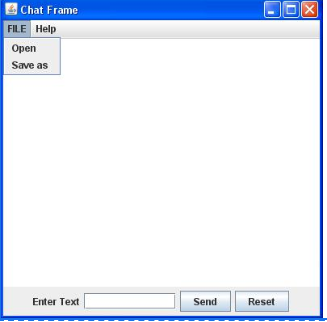
}

}

scanner.close();

}

}

9. 

import javax.swing.\*;

import java.awt.\*;

class gui {

public static void main(String args[]) {

//Create the Frame

JFrame jframe = new JFrame("Chat Screen");

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

jframe.setSize(800, 500);

// create two menubar button FILE and HELP

JMenuBar menuBar = new JMenuBar();

JMenu fileMenu = new JMenu("FILE");

JMenu helpMenu = new JMenu("Help");

menuBar.add(fileMenu);

menuBar.add(helpMenu);

// create two more option in FILE button

JMenuItem fileMenu1 = new JMenuItem("Open");

JMenuItem fileMenu2 = new JMenuItem("Save as");

fileMenu.add(fileMenu1);

fileMenu.add(fileMenu2);

// Text Area at the Center

JTextArea textArea = new JTextArea();

//Create the panel at bottom and add label, textArea and buttons

JPanel panel = new JPanel(); // this panel is not visible in output

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

JTextField textField = new JTextField(15); // accepts upto 15 characters

JButton btn\_send = new JButton("Send");

JButton btn\_reset = new JButton("Reset");

panel.add(label); // Components Added using Flow Layout

panel.add(textField);

panel.add(btn\_send);

panel.add(btn\_reset);

//Adding Components to the frame.

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

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

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

jframe.setVisible(true);

}

}

10. 

import javax.swing.\*;

import java.awt.\*;

public class NumberAddition2 {

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);

}

}

11.

11.1) Write a Java program that takes a number as input and prints its multiplication table up to 10. Test Data:  
Input a number: 8  
Expected Output :  
8 x 1 = 8  
8 x 2 = 16  
8 x 3 = 24  
...  
8 x 10 = 80

import java.util.Scanner;

public class MultiplicationTable {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

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 Prime {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

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

int num;

num=sc.nextInt();

boolean flag = false;

for (int i = 2; i <= num / 2; ++i) {

// condition for nonprime number

if (num % i == 0) {

flag = true;

break;

}

}

if (!flag)

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

else

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

}

}

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

import java.util.Scanner;

public class DiamondPattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int rows = scanner.nextInt();

// 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 MatrixTransposeExample {

public static void main(String args[]) {

// Creating a matrix

int original[][] = new int[3][3];

Scanner scanner = new Scanner(System.in);

// Taking user input for the original matrix

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

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

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

}

}

// Creating another matrix to store the transpose of the matrix

int transpose[][] = new int[3][3]; // 3 rows and 3 columns

// Code to transpose a matrix

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

transpose[i][j] = original[j][i];

}

}

System.out.println("Printing Matrix without transpose:");

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

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

}

System.out.println(); // new line

}

System.out.println("Printing Matrix After Transpose:");

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

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

}

System.out.println(); // new line

}

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 WordCount {

public static void main(String[] args) {

try {

File file = new File("path/to/your/file.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 javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Calculator extends JFrame implements ActionListener {

private JTextField textField;

private JButton[] numberButtons;

private JButton[] operatorButtons;

private JButton equalsButton;

private JButton clearButton;

private JPanel panel;

private String firstNumber;

private String operator;

private boolean isResultShown;

public Calculator() {

setTitle("Calculator");

setSize(300, 400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

textField = new JTextField();

textField.setPreferredSize(new Dimension(280, 50));

textField.setEditable(false);

textField.setHorizontalAlignment(JTextField.RIGHT);

numberButtons = new JButton[10];

operatorButtons = new JButton[4];

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

numberButtons[i] = new JButton(String.valueOf(i));

}

operatorButtons[0] = new JButton("+");

operatorButtons[1] = new JButton("-");

operatorButtons[2] = new JButton("\*");

operatorButtons[3] = new JButton("/");

equalsButton = new JButton("=");

clearButton = new JButton("C");

panel = new JPanel(new GridLayout(4, 4));

panel.add(numberButtons[7]);

panel.add(numberButtons[8]);

panel.add(numberButtons[9]);

panel.add(operatorButtons[0]);

panel.add(numberButtons[4]);

panel.add(numberButtons[5]);

panel.add(numberButtons[6]);

panel.add(operatorButtons[1]);

panel.add(numberButtons[1]);

panel.add(numberButtons[2]);

panel.add(numberButtons[3]);

panel.add(operatorButtons[2]);

panel.add(numberButtons[0]);

panel.add(clearButton);

panel.add(equalsButton);

panel.add(operatorButtons[3]);

setLayout(new BorderLayout());

add(textField, BorderLayout.NORTH);

add(panel, BorderLayout.CENTER);

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

numberButtons[i].addActionListener(this);

}

for (int i = 0; i < 4; i++) {

operatorButtons[i].addActionListener(this);

}

equalsButton.addActionListener(this);

clearButton.addActionListener(this);

}

public void actionPerformed(ActionEvent e) {

Object source = e.getSource();

if (source == clearButton) {

textField.setText("");

firstNumber = "";

operator = "";

} else if (source == equalsButton) {

String secondNumber = textField.getText();

double result = calculateResult(Double.parseDouble(firstNumber), Double.parseDouble(secondNumber), operator);

textField.setText(String.valueOf(result));

isResultShown = true;

} else {

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

if (source == numberButtons[i]) {

if (isResultShown) {

textField.setText("");

isResultShown = false;

}

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

break;

}

}

for (int i = 0; i < 4; i++) {

if (source == operatorButtons[i]) {

firstNumber = textField.getText();

operator = operatorButtons[i].getText();

textField.setText("");

break;

}

}

}

}

private double calculateResult(double firstNumber, double secondNumber, String operator) {

double result = 0;

switch (operator) {

case "+":

result = firstNumber + secondNumber;

break;

case "-":

result = firstNumber - secondNumber;

break;

case "\*":

result = firstNumber \* secondNumber;

break;

case "/":

result = firstNumber / secondNumber;

break;

}

return result;

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

public void run() {

new 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

import java.util.ArrayList;

import java.util.Iterator;

public class ArrayListIteration {

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 WordCountHash {

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 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("\nCombined strings:");

String combinedString = combineStrings(strings);

System.out.println(combinedString);

System.out.println("\nReversed first string:");

String reversedString = reverseString(strings[0]);

System.out.println(reversedString);

scanner.close();

}

public static void sortStrings(String[] strings) {

Arrays.sort(strings);

for (String str : strings) {

System.out.println(str);

}

}

public static String combineStrings(String[] strings) {

StringBuilder combined = new StringBuilder();

for (String str : strings) {

combined.append(str);

}

return combined.toString();

}

public static String reverseString(String str) {

return new StringBuilder(str).reverse().toString();

}

}

18. Write a program to implement following inheritance. Accept data for 5 persons and display the name of employee having salary greater than 5000.  
  
Class Name: Person  
Member variables:  
Name, age  
  
Class Name: Employee  
Member variables:  
Designation, salary

import java.util.Scanner;

class Person {

protected String name;

protected int age;

public void 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 EmployeeProgram {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Employee[] employees = new Employee[5];

for (int i = 0; i < 5; 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.

interface Account {

void set(String accountNumber, double balance);

void display();

}

interface Person {

void store(String name);

void disp();

}

class Customer implements Person, Account {

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);

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

}

@Override

public void store(String name) {

this.name = name;

}

@Override

public void disp() {

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

}

private double calculateInterest() {

// Calculate interest based on balance

return balance \* 0.05; // Assuming 5% interest rate

}

}

public class MultipleInheritanceExample {

public static void main(String[] args) {

Customer customer = new Customer();

// Set account details

customer.set("123456789", 10000);

// Set person details

customer.store("John Doe");

// Display account and person information

customer.disp();

customer.display();

}

}

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("Video Title: " + title);

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

}

}

class Movie extends VideoTape {

private String director;

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

super(title, duration);

this.director = director;

}

public String getDirector() {

return director;

}

@Override

public void display() {

super.display();

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

}

}

class MusicVideo extends VideoTape {

private String artist;

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

super(title, duration);

this.artist = artist;

}

public String getArtist() {

return artist;

}

@Override

public void display() {

super.display();

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

}

}

public class HierarchyExample {

public static void main(String[] args) {

VideoTape video = new VideoTape("Generic Video", 120);

video.display();

System.out.println();

Movie movie = new Movie("Inception", 150, "Christopher Nolan");

movie.display();

System.out.println();

MusicVideo musicVideo = new MusicVideo("Shape of You", 180, "Ed Sheeran");

musicVideo.display();

}

}

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

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

class Student {

private String name;

private int grade;

private List<String> courses;

public Student(String name, int grade) {

this.name = name;

this.grade = grade;

this.courses = new ArrayList<>();

}

public void addCourse(String course) {

courses.add(course);

}

public void removeCourse(String course) {

courses.remove(course);

}

public void displayCourses() {

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

for (String course : courses) {

System.out.println("- " + course);

}

}

}

public class StudentProgram {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String name = scanner.nextLine();

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

int grade = scanner.nextInt();

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

Student student = new Student(name, grade);

char choice;

do {

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

System.out.println("1. Add course");

System.out.println("2. Remove course");

System.out.println("3. Display courses");

System.out.println("4. Exit");

System.out.print("Enter your choice (1-4): ");

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

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

switch (choice) {

case '1':

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

String courseToAdd = scanner.nextLine();

student.addCourse(courseToAdd);

break;

case '2':

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

String courseToRemove = scanner.nextLine();

student.removeCourse(courseToRemove);

break;

case '3':

student.displayCourses();

break;

case '4':

System.out.println("Exiting program...");

break;

default:

System.out.println("Invalid choice! Please try again.");

}

} while (choice != '4');

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.

class Person {

private String firstName;

private String lastName;

public Person(String firstName, String lastName) {

this.firstName = firstName;

this.lastName = lastName;

}

public String getFirstName() {

return firstName;

}

public String getLastName() {

return lastName;

}

}

class Employee extends Person {

private int employeeId;

private String jobTitle;

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

super(firstName, lastName);

this.employeeId = employeeId;

this.jobTitle = jobTitle;

}

public int getEmployeeId() {

return employeeId;

}

@Override

public String getLastName() {

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

}

}

public class PersonEmployeeProgram {

public static void main(String[] args) {

Person person = new Person("John", "Doe");

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

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

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

System.out.println();

Employee employee = new Employee("Jane", "Smith", 12345, "Manager");

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

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

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

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

}

}

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

import java.util.HashSet;

public class LongestConsecutiveSequence {

public static int longestConsecutive(int[] nums) {

// Create a set to store all the elements in the array

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

for (int num : nums) {

set.add(num);

}

int maxLength = 0;

// Iterate over each element in the array

for (int num : nums) {

// Check if the element is the start of a sequence

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

int currentNum = num;

int currentLength = 1;

// Calculate the length of the consecutive sequence

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

currentNum++;

currentLength++;

}

// Update the maximum length if necessary

maxLength = Math.max(maxLength, currentLength);

}

}

return maxLength;

}

public static void main(String[] args) {

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

int length = longestConsecutive(nums);

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

}

}

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

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 {

this.rollNo = rollNo;

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

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

}

this.age = age;

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

throw new NameNotValidException("Name is not valid. It should only contain alphabets and spaces.");

}

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 StudentTest {

public static void main(String[] args) {

try {

Student student1 = new Student(1, "John Doe", 20, "Computer Science");

System.out.println("Student 1:");

System.out.println("Roll No: " + student1.getRollNo());

System.out.println("Name: " + student1.getName());

System.out.println("Age: " + student1.getAge());

System.out.println("Course: " + student1.getCourse());

Student student2 = new Student(2, "Jane123", 22, "Electrical Engineering");

System.out.println("Student 2:");

System.out.println("Roll No: " + student2.getRollNo());

System.out.println("Name: " + student2.getName());

System.out.println("Age: " + student2.getAge());

System.out.println("Course: " + student2.getCourse());

} catch (AgeNotWithinRangeException e) {

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

} catch (NameNotValidException e) {

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

}

}

}

25. 

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();

}

}

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();

}

}

}