1.1) Program to remove all repeated elements from an array -done

import java.util.Arrays;  
import java.util.Scanner;  
  
public class remove\_repeat\_ele {  
 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[] arr = new int[size];  
 System.*out*.println("Enter the elements of the array:");  
 for (int i = 0; i < size; i++) {  
 arr[i] = scanner.nextInt();  
 }  
  
 // Sort the array  
 Arrays.*sort*(arr);  
  
 // Find the count of unique elements  
 int uniqueCount = 0;  
 for (int i = 0; i < arr.length - 1; i++) {  
 if (arr[i] != arr[i + 1]) {  
 uniqueCount++;  
 }  
 }  
 uniqueCount++;  
  
 // Create a new array with unique elements  
 int[] result = new int[uniqueCount];  
 int index = 0;  
 for (int i = 0; i < arr.length - 1; i++) {  
 if (arr[i] != arr[i + 1]) {  
 result[index++] = arr[i];  
 }  
 }  
// This is for last ele  
 result[index] = arr[arr.length - 1];  
  
 // Print the array without duplicates  
 System.*out*.println("Array without duplicates: " + Arrays.*toString*(result));  
 }  
}

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

import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.Scanner;  
  
public class common\_ele {  
 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[] arr1 = new int[size1];  
 System.*out*.println("Enter the elements of the first array:");  
 for (int i = 0; i < size1; i++) {  
 arr1[i] = scanner.nextInt();  
 }  
  
 System.*out*.print("Enter the size of the second array: ");  
 int size2 = scanner.nextInt();  
  
 int[] arr2 = new int[size2];  
 System.*out*.println("Enter the elements of the second array:");  
 for (int i = 0; i < size2; i++) {  
 arr2[i] = scanner.nextInt();  
 }  
  
 ArrayList<Integer> commonElements = new ArrayList<>();  
  
 // Sort the arrays  
 Arrays.*sort*(arr1);  
 Arrays.*sort*(arr2);  
  
 int i = 0;  
 int j = 0;  
  
 while (i < arr1.length && j < arr2.length) {  
 if (arr1[i] == arr2[j]) {  
 commonElements.add(arr1[i]);  
 i++;  
 j++;  
 } else if (arr1[i] < arr2[j]) {  
 i++;  
 } else {  
 j++;  
 }  
 }  
  
 // Convert ArrayList to array  
 int[] result = new int[commonElements.size()];  
 for (int k = 0; k < commonElements.size(); k++) {  
 result[k] = commonElements.get(k);  
 }  
  
 System.*out*.println("Common elements: " + Arrays.*toString*(result));  
 }  
}

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

import java.util.Arrays;  
import java.util.Scanner;  
  
public class CountDuplicateWords {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Enter a string: ");  
 String input = scanner.nextLine();  
  
 // Split the input string into words  
 String[] words = input.toLowerCase().split("\\s+");  
  
 // Sort the array of words  
 Arrays.*sort*(words);  
  
 // Count the number of duplicate words  
 int duplicateCount = 0;  
 for (int i = 0; i < words.length - 1; i++) {  
 if (words[i].equals(words[i + 1])) {  
 duplicateCount++;  
 // Skip to the next unique word  
// This ensures that all consecutive occurrences of the same word are skipped.  
 while (i < words.length - 1 && words[i].equals(words[i + 1])) {  
 i++;  
 }  
 }  
 }  
  
 System.*out*.println("Number of duplicate words: " + duplicateCount);  
 }  
}

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

public class umbrella {  
 public static void main(String[] args) {  
 String word = "umbrella";  
 char targetChar = 'e';  
  
 // Check if the string contains the target character  
 boolean containsTargetChar = word.contains(String.*valueOf*(targetChar));  
  
 // Print the result  
 if (containsTargetChar) {  
 System.*out*.println("The string contains the character '" + targetChar + "'.");  
 } else {  
 System.*out*.println("The string does not contain the character '" + targetChar + "'.");  
 }  
 }  
}

3.

3.1)Java Program to Reverse a String. -done

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 str = scanner.nextLine();  
  
 String reversedStr = *reverseString*(str);  
 System.*out*.println("Reversed String: " + reversedStr);  
 }  
  
 public static String reverseString(String str) {  
 StringBuilder reversed = new StringBuilder();  
 for (int i = str.length() - 1; i >= 0; i--) {  
 reversed.append(str.charAt(i));  
 }  
 return reversed.toString();  
 }  
}

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

import java.util.Scanner;  
  
public class PalindromeCheck {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Enter a string: ");  
 String str = scanner.nextLine();  
  
 boolean isPalindrome = *checkPalindrome*(str);  
  
 if (isPalindrome) {  
 System.*out*.println("The string is a palindrome.");  
 } else {  
 System.*out*.println("The string is not a palindrome.");  
 }  
 }  
  
 public static boolean checkPalindrome(String str) {  
 int left = 0;  
 int 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.-done

import java.util.Scanner;  
  
interface Vehicle {  
 void move();  
}  
  
class Helicopter implements Vehicle {  
 @Override  
 public void move() {  
 System.*out*.println("The Helicopter flies in the air.");  
 }  
}  
  
class Car implements Vehicle {  
 @Override  
 public void move() {  
 System.*out*.println("The Car drives on the road.");  
 }  
}  
  
class Train implements Vehicle {  
 @Override  
 public void move() {  
 System.*out*.println("The 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, or Train): ");  
 String vehicleType = scanner.nextLine();  
  
 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. Please choose from Helicopter, Car, or Train.");  
 scanner.close();  
 return;  
 }  
  
 vehicle.move();  
  
 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 eac of the two classes and print the percentage of marks for both the students.-done-

import java.util.Scanner;  
  
abstract class Marks {  
 public abstract double getPercentage();  
}  
  
class A extends Marks {  
 private int subject1;  
 private int subject2;  
 private int subject3;  
  
 public A(int subject1, int subject2, int subject3) {  
 this.subject1 = subject1;  
 this.subject2 = subject2;  
 this.subject3 = subject3;  
 }  
  
 public double getPercentage() {  
 double totalMarks = subject1 + subject2 + subject3;  
 return (totalMarks / 300) \* 100;  
 }  
}  
  
class B extends Marks {  
 private int subject1;  
 private int subject2;  
 private int subject3;  
 private int subject4;  
  
 public B(int subject1, int subject2, int subject3, int subject4) {  
 this.subject1 = subject1;  
 this.subject2 = subject2;  
 this.subject3 = subject3;  
 this.subject4 = subject4;  
 }  
  
 public double getPercentage() {  
 double totalMarks = subject1 + subject2 + subject3 + subject4;  
 return (totalMarks / 400) \* 100;  
 }  
}  
  
public class Percentage {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.println("Enter marks for Student A (in three subjects):");  
 int subject1A = scanner.nextInt();  
 int subject2A = scanner.nextInt();  
 int subject3A = scanner.nextInt();  
  
 System.*out*.println("Enter marks for Student B (in four subjects):");  
 int subject1B = scanner.nextInt();  
 int subject2B = scanner.nextInt();  
 int subject3B = scanner.nextInt();  
 int subject4B = scanner.nextInt();  
  
 A studentA = new A(subject1A, subject2A, subject3A);  
 B studentB = new B(subject1B, subject2B, subject3B, subject4B);  
  
 System.*out*.println("Percentage of marks for Student A: " + studentA.getPercentage() + "%");  
 System.*out*.println("Percentage of marks for Student B: " + studentB.getPercentage() + "%");  
  
 scanner.close();  
 }  
}

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

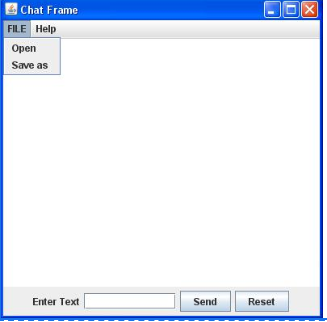
import java.io.\*;  
import java.util.\*;  
import java.text.\*;  
import java.math.\*;  
import java.util.regex.\*;  
//Write your code here  
class Arithmetic{  
 public int add(int a, int b){  
 int sum = a + b;  
 return sum;  
 }  
}  
class Adder extends Arithmetic{  
 public int callAdd(int a, int b){  
 return add(a, b);  
 }  
}  
class 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");  
 }  
}

**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”.-ntr

import java.util.Scanner;  
class MyCalculator {  
 /\*  
 \* Create the method long power(int, int) here.  
 \*/  
 public long power(int n, int p) throws Exception  
 {  
 if(n == 0 && p == 0)  
 throw new Exception("n and p should not be zero.");  
 else if(n < 0 || p < 0)  
 throw new Exception("n or p should not be negative.");  
 else  
 return (long)(Math.*pow*(n,p));  
 }  
}  
class Solution {  
 public static final MyCalculator *my\_calculator* = new MyCalculator();  
 public static final Scanner *in* = new Scanner(System.*in*);  
 public static void main(String[] args) {  
 while (*in* .hasNextInt()) {  
 int n = *in* .nextInt();  
 int p = *in* .nextInt();  
 try {  
 System.*out*.println(*my\_calculator*.power(n, p));  
 } catch (Exception e) {  
 System.*out*.println(e);  
 }  
 }  
 }  
}

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.-ntr

import java.util.HashMap;  
import java.util.Map;  
import java.util.Scanner;  
public class Solution {  
 /\* input Scanner for reading input for the program \*/  
 private static Scanner *inputScanner*;  
 public static void main(String[] args) {  
 /\* Create a phone book Map \*/  
 Map<String, String> phoneBook = new HashMap<String, String>();  
 /\* Create the input Scanner instance \*/  
 *inputScanner* = new Scanner(System.*in*);  
 /\* Read the number of input phone numbers \*/  
 int numFriends = *inputScanner*.nextInt();  
 /\* Go to the next line of input \*/  
 *inputScanner*.nextLine();  
 /\* Loop thru the number of friends, reading the name and phone number, and adding to the phoneBook \*/  
 for (int i = 0; i < numFriends; i++) {  
 /\* Read the name of the friend \*/  
 String name = *inputScanner*.nextLine();  
 /\* Read the phone number \*/  
 String phone = *inputScanner*.nextLine();  
 /\* Put the name and phone number into the phoneBook \*/  
 phoneBook.put(name, phone);  
 }  
 /\* Loop while there is still more input data \*/  
 while (*inputScanner*.hasNext()) {  
 /\* Read the person to search for a phone number \*/  
 String inputName = *inputScanner*.nextLine();  
 /\* Does the name exist in the phone Book \*/  
 if (phoneBook.containsKey(inputName)) {  
 /\* Yes, print out the name and phone number \*/  
 System.*out*.println(inputName + "=" + phoneBook.get(inputName));  
 } else {  
 /\* Print out error message if not found \*/  
 System.*out*.println("Not found");  
 }  
 }  
 }  
}

9. 

import javax.swing.\*;  
import java.awt.\*;  
class gui {  
 public static void main(String args[]) {  
 //Creating the Frame  
 JFrame frame = new JFrame("Chat Frame");  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 frame.setSize(400, 400);  
 //Creating the MenuBar and adding components  
 JMenuBar mb = new JMenuBar();  
 JMenu m1 = new JMenu("FILE");  
 JMenu m2 = new JMenu("Help");  
 mb.add(m1);  
 mb.add(m2);  
 JMenuItem m11 = new JMenuItem("Open");  
 JMenuItem m22 = new JMenuItem("Save as");  
 m1.add(m11);  
 m1.add(m22);  
 //Creating the panel at bottom and adding components  
 JPanel panel = new JPanel(); // the panel is not visible in output  
 JLabel label = new JLabel("Enter Text");  
 JTextField tf = new JTextField(10); // accepts upto 10 characters  
 JButton send = new JButton("Send");  
 JButton reset = new JButton("Reset");  
 panel.add(label); // Components Added using Flow Layout  
 panel.add(tf);  
 panel.add(send);  
 panel.add(reset);  
 // Text Area at the Center  
 JTextArea ta = new JTextArea();  
 //Adding Components to the frame.  
 frame.getContentPane().add(BorderLayout.*SOUTH*, panel);  
 frame.getContentPane().add(BorderLayout.*NORTH*, mb);  
 frame.getContentPane().add(BorderLayout.*CENTER*, ta);  
 frame.setVisible(true);  
 }  
}

10. -done revise2

import javax.swing.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class Question10 {  
  
 public static void main(String[] args) {  
  
 JFrame f1= new JFrame();  
 f1.setTitle("Number Addition");  
  
 JLabel firstno= new JLabel("First : ");  
 firstno.setBounds(40,50,40,60);  
  
 JLabel Secondno= new JLabel("Second: ");  
 Secondno.setBounds(40,100,80,60);  
  
 JLabel Result= new JLabel("Result: ");  
 Result.setBounds(40,150,80,60);  
  
  
 JTextField t1= new JTextField();  
 t1.setBounds(100,60,200,30);  
  
 JTextField t2= new JTextField();  
 t2.setBounds(100,120,200,30);  
  
 JTextField ResultField= new JTextField();  
 ResultField.setBounds(100,170,200,30);  
  
  
 JButton Add = new JButton("Add");  
 Add.setBounds(100,230,80,30);  
  
 Add.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
  
 String firstno1= t1.getText();  
 int numberFirst=Integer.*valueOf*(firstno1);  
  
  
 String secondno1=t2.getText();  
 int numberSecond=Integer.*valueOf*(secondno1);  
  
  
 int Result=numberSecond+numberFirst;  
 String ResultString=String.*valueOf*(Result);  
  
 ResultField.setText(ResultString);  
  
 }  
 });  
  
 JButton Clear = new JButton("Clear");  
 Clear.setBounds(250,230,80,30);  
 Clear.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 t1.setText(" ");  
 t2.setText(" ");  
 ResultField.setText(" ");  
 }  
 });  
  
 JButton Exit= new JButton("Exit");  
 Exit.setBounds(350,350,80,30);  
 Exit.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 f1.dispose();  
 }  
 });  
  
  
  
 f1.setLayout(null);  
  
 f1.add(firstno);  
 f1.add(Secondno);  
 f1.add(t1);  
 f1.add(t2);  
 f1.add(Result);  
 f1.add(ResultField);  
 f1.add(Add);  
 f1.add(Clear);  
 f1.add(Exit);  
  
  
 f1.setSize(500,500);  
  
 f1.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 -done

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

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

import java.util.Scanner;  
  
class Prime {  
 public static void main(String[] args) {  
 Scanner sc= new Scanner(System.*in*);  
 System.*out*.println("Enter a number: ");  
 int number= sc.nextInt();  
 if(*isPrime*(number)) {  
 System.*out*.println(number + " is prime number");  
 }  
 else{  
 System.*out*.println(number + " is a non-prime number");  
 }  
 }  
 static boolean isPrime(int num)  
 {  
 if(num<=1)  
 {  
 return false;  
 }  
 for(int i=2;i<=num/2;i++)  
 {  
 if((num%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 :-done  
  
  
\*   
\*\*\*   
\*\*\*\*\*   
\*\*\*\*\*\*\*   
\*\*\*\*\*\*\*\*\*   
\*\*\*\*\*\*\*\*\*\*\*   
\*\*\*\*\*\*\*\*\*\*\*\*\*   
\*\*\*\*\*\*\*\*\*\*\*   
\*\*\*\*\*\*\*\*\*   
\*\*\*\*\*\*\*   
\*\*\*\*\*   
\*\*\*   
\*

import java.util.Scanner;  
public class DiamondPattern {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter the number of rows: ");  
 int n = sc.nextInt();  
 int i;  
 int j;  
 for (i=1;i<=n;i++){  
 for (j=1;j<=i;j++){  
 System.*out*.print("\* ");  
 }  
 System.*out*.println();  
 }  
 for (i=1;i<=n-1;i++){  
 for (j=n-1;j>=i;j--){  
 System.*out*.print("\* ");  
 }  
 System.*out*.println();  
 }  
 }  
}

13.

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

import java.util.Scanner;  
public class TransposeMatrix  
{  
 public static void main(String args[])  
 {  
 int i, j;  
 System.*out*.println("Enter total rows and columns: ");  
 Scanner s = new Scanner(System.*in*);  
 int row = s.nextInt();  
 int column = s.nextInt();  
 int array[][] = new int[row][column];  
 System.*out*.println("Enter matrix:");  
 for(i = 0; i < row; i++)  
 {  
 for(j = 0; j < column; j++)  
 {  
 array[i][j] = s.nextInt();  
 System.*out*.print("");  
 }  
 }  
 System.*out*.println("The above matrix before Transpose is ");  
 for(i = 0; i < row; i++)  
 {  
 for(j = 0; j < column; j++)  
 {  
 System.*out*.print(array[i][j]+" ");  
 }  
 System.*out*.println(" ");  
 }  
 System.*out*.println("The above matrix after Transpose is ");  
 for(i = 0; i < column; i++)  
 {  
 for(j = 0; j < row; j++)  
 {  
 System.*out*.print(array[j][i]+" ");  
 }  
 System.*out*.println(" ");  
 }  
 }  
}

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

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("C:\\Users\\Tilak\\Documents\\hello.txt"); // Replace "input.txt" with the path to your text file  
 Scanner scanner = new Scanner(file);  
  
 int wordCount = 0;  
  
 while (scanner.hasNext()) {  
 String line = scanner.nextLine();  
 String[] words = line.split(" ");  
 wordCount += words.length;  
 }  
  
 System.*out*.println("Number of words in the file: " + wordCount);  
  
 scanner.close();  
 } catch (FileNotFoundException e) {  
 System.*out*.println("File not found.");  
 }  
 }  
}

14. -ntr-revise

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.util.ArrayList;  
import java.util.Arrays;  
  
public class Question14 {  
  
 public static void main(String[] args) {  
 JFrame f1= new JFrame();  
 f1.setTitle("Swing Calculator");  
  
 JTextField textField =new JTextField();  
 textField.setBounds(60,50,300,50);  
 textField.setText("");  
  
 JPanel panel= new JPanel();  
 panel.setLayout(new GridLayout(4,4));  
 panel.setBounds(60,130,300,50);  
  
 JButton b7=new JButton("7");  
 b7.setBounds(60,150,100,40);  
 panel.add(b7);  
 b7.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String exp=textField.getText();  
 String exp2=exp.concat("7");  
 textField.setText(exp2);  
 }  
 });  
  
 JButton b8=new JButton("8");  
 b8.setBounds(66,150,100,40);  
 b8.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String exp=textField.getText();  
 String exp2=exp.concat("8");  
 textField.setText(exp2);  
 }  
 });  
 panel.add(b8);  
  
 JButton b9=new JButton("9");  
 b9.setBounds(72,150,100,40);  
 b9.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String exp=textField.getText();  
 String exp2=exp.concat("9");  
 textField.setText(exp2);  
 }  
 });  
 panel.add(b9);  
  
 JButton badd=new JButton("+");  
 badd.setBounds(78,150,100,40);  
 badd.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String field= textField.getText();  
 String new1=field.concat("+");  
 textField.setText(new1);  
 }  
 });  
 panel.add(badd);  
  
 JButton b4=new JButton("4");  
 b4.setBounds(60,170,100,40);  
 b4.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String exp=textField.getText();  
 String exp2=exp.concat("4");  
 textField.setText(exp2);  
 }  
 });  
 panel.add(b4);  
  
 JButton b5=new JButton("5");  
 b5.setBounds(66,170,100,40);  
 b5.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String exp=textField.getText();  
 String exp2=exp.concat("5");  
 textField.setText(exp2);  
 }  
 });  
 panel.add(b5);  
  
 JButton b6=new JButton("6");  
 b6.setBounds(72,170,100,40);  
 b6.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String exp=textField.getText();  
 String exp2=exp.concat("6");  
 textField.setText(exp2);  
 }  
 });  
 panel.add(b6);  
  
 JButton bsub=new JButton("-");  
 bsub.setBounds(78,170,100,40);  
 panel.add(bsub);  
  
 JButton b1=new JButton("1");  
 b1.setBounds(60,190,100,40);  
 b1.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String exp=textField.getText();  
 String exp2=exp.concat("1");  
 textField.setText(exp2);  
 }  
 });  
  
 panel.add(b1);  
  
 JButton b2=new JButton("2");  
 b2.setBounds(66,170,100,40);  
 b2.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String exp=textField.getText();  
 String exp2=exp.concat("2");  
 textField.setText(exp2);  
 }  
 });  
 panel.add(b2);  
  
 JButton b3=new JButton("3");  
 b3.setBounds(72,170,100,40);  
 panel.add(b3);  
 b3.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String exp=textField.getText();  
 String exp2=exp.concat("3");  
 textField.setText(exp2);  
 }  
 });  
  
 JButton bmul=new JButton("\*");  
 bmul.setBounds(78,170,100,40);  
 panel.add(bmul);  
  
 JButton bcancel=new JButton("C");  
 bcancel.setBounds(60,190,100,40);  
 bcancel.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 textField.setText("");  
 }  
 });  
 panel.add(bcancel);  
  
 JButton b0=new JButton("0");  
 b0.setBounds(66,170,100,40);  
 b0.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String exp=textField.getText();  
 String exp2=exp.concat("0");  
 textField.setText(exp2);  
 }  
 });  
 panel.add(b0);  
  
 JButton bequals=new JButton("=");  
 bequals.setBounds(72,170,100,40);  
 panel.add(bequals);  
 bequals.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String Expression= textField.getText();  
  
 if (Expression.contains("+")){  
 String newexp1=Expression.replace("+"," ");  
 String[] newexp=newexp1.split(" ");  
 int size=newexp.length;  
 int arrayofnewxexp[]=new int[size];  
 int sum=0;  
 for (int i=0;i<size;i++){  
 arrayofnewxexp[i]=Integer.*parseInt*(newexp[i]);  
 }  
 for (int i=0;i<size;i++){  
 sum=sum+arrayofnewxexp[i];  
 }  
 textField.setText(String.*valueOf*(sum));  
 }  
  
 }  
 });  
  
 JButton bdiv=new JButton("/");  
 bdiv.setBounds(78,170,100,40);  
 panel.add(bdiv);  
  
 f1.setVisible(true);  
 f1.add(textField,BorderLayout.*NORTH*);  
 f1.add(panel);  
 f1.setLayout(null);  
 f1.setSize(500,500);  
 }  
}

15. Write a Java Program to iterate ArrayList using for-loop, iterator, and advance for-loop. Insert 3 Array List.Input 20 30 40Output:-done  
  
  
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);  
  
 // Iterator Loop  
 System.*out*.println("Iterator Loop:");  
 Iterator<Integer> iterator = numbers.iterator();  
// checks if there are any more elements in the collection to be iterated over. If there are elements remaining, the loop continues; otherwise, it exits.  
 while (iterator.hasNext()) {  
 System.*out*.println(iterator.next());  
 }  
  
 // Advanced For Loop  
 System.*out*.println("Advanced For Loop:");  
 for (int num : numbers) {  
 System.*out*.println(num);  
 }  
  
 // For Loop  
 System.*out*.println("For Loop:");  
 for (int i = 0; i < numbers.size(); i++) {  
 System.*out*.println(numbers.get(i));  
 }  
 }  
}  
//can we take input from user

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}-done hash map vid- revise also

import java.util.HashMap;  
import java.util.Map;  
public class WordCount\_hashMap {  
 public static void main(String[] args) {  
 String str = "This this is is done by Saket Saket";  
 Map<String, Integer> hashMap = new HashMap<>();  
 String[] words = str.split(" ");  
 for (String word : words) {  
 Integer integer = hashMap.get(word);  
 if (integer == null)  
 hashMap.put(word, 1);  
 else {  
 hashMap.put(word, integer + 1);  
 }  
 }  
 System.*out*.println(hashMap);  
 }  
}

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 .-done-

import java.util.Arrays;  
import java.util.Scanner;  
  
public class StringOperation {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 // Read 10 strings from the console  
 String[] strings = new String[10];  
 System.*out*.println("Enter 10 strings:");  
 for (int i = 0; i < 10; i++) {  
 strings[i] = scanner.nextLine();  
 }  
  
 // Sort the strings  
 Arrays.*sort*(strings);  
  
 // Print the sorted strings  
 System.*out*.println("Sorted strings:");  
 for (String str : strings) {  
 System.*out*.println(str);  
 }  
  
 // Combine two strings  
 String combinedString = strings[0] + strings[1];  
 System.*out*.println("Combined string: " + combinedString);  
  
 // Reverse the first string  
 String reversedString = *reverseString*(strings[0]);  
 System.*out*.println("Reversed first string: " + reversedString);  
 }  
  
 // Method to reverse a string  
 public static String reverseString(String str) {  
 StringBuilder reversed = new StringBuilder();  
 for (int i = str.length() - 1; i >= 0; i--) {  
 reversed.append(str.charAt(i));  
 }  
 return reversed.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.-done-revise  
  
Class Name: Person  
Member variables:  
Name, age  
  
Class Name: Employee  
Member variables:  
Designation, salary

import java.util.Scanner;  
  
class Person {  
 protected String name;  
 protected int age;  
  
 public void acceptData() {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter name: ");  
 name = scanner.nextLine();  
 System.*out*.print("Enter age: ");  
 age = scanner.nextInt();  
 }  
}  
  
class Employee extends Person {  
 private String designation;  
 private double salary;  
  
 public void acceptData() {  
 super.acceptData();  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter designation: ");  
 designation = scanner.nextLine();  
 System.*out*.print("Enter salary: ");  
 salary = scanner.nextDouble();  
 }  
  
 public double getSalary() {  
 return salary;  
 }  
  
 public void displayData() {  
 System.*out*.println("Name: " + name);  
 System.*out*.println("Age: " + age);  
 System.*out*.println("Designation: " + designation);  
 System.*out*.println("Salary: " + salary);  
 }  
}  
  
public class InheritanceExample {  
 public static void main(String[] args) {  
 Employee[] employees = new Employee[5];  
  
 System.*out*.println("Enter data for 5 employees:");  
 for (int i = 0; i < employees.length; i++) {  
 employees[i] = new Employee();  
 System.*out*.println("Employee " + (i + 1) + ":");  
 employees[i].acceptData();  
 }  
  
 System.*out*.println("\nEmployees with salary greater than 5000:");  
 for (int i = 0; i < employees.length; i++) {  
 if (employees[i].getSalary() > 5000) {  
 System.*out*.println(employees[i].name);  
 }  
 }  
 }  
}

19. Implementing “Multiple Inheritance”. Create a two interfaces Account containing methods set() and display() And interface Person containing methods store() and disp(). Derive a class Customer from Person and Account. Accept the name, account number, balance and display all the information related to account along with the interest. Revise22

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 MultipleInheritanceExample {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 Customer customer = new Customer();  
 System.*out*.print("Enter name: ");  
 String name = scanner.nextLine();  
 customer.store(name);  
 System.*out*.print("Enter account number: ");  
 String accountNumber = scanner.nextLine();  
 System.*out*.print("Enter balance: $");  
 double balance = scanner.nextDouble();  
 scanner.nextLine();  
 customer.set(accountNumber, balance);  
 System.*out*.println("\nCustomer Information:");  
 customer.disp();  
 customer.display();  
 System.*out*.print("\nEnter interest rate (%): ");  
 double interestRate = scanner.nextDouble();  
 System.*out*.println("\nAccount Information with Interest:");  
 customer.calculateInterest(interestRate);  
 scanner.close();  
 }  
}

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



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.-ntr-revise

import java.util.ArrayList;  
import java.util.List;  
public class Student {  
 private String name;  
 private int grade;  
 private List<String> courses;  
 public Student(String name, int grade) {  
 this.name = name;  
 this.grade = grade;  
 this.courses = new ArrayList<>();  
 }  
 public String getName() {  
 return name;  
 }  
 public int getGrade() {  
 return grade;  
 }  
 public void addCourse(String course) {  
 courses.add(course);  
 }  
 public void removeCourse(String course) {  
 courses.remove(course);  
 }  
 public List<String> getCourses() {  
 return courses;  
 }  
 public static void main(String[] args) {  
 // Creating a student object  
 Student student = new Student("John Doe", 10);  
 // Adding courses  
 student.addCourse("Math");  
 student.addCourse("Science");  
 student.addCourse("English");  
 // Removing a course  
 student.removeCourse("Science");  
 // Getting the student's name, grade, and courses  
 System.*out*.println("Student Name: " + student.getName());  
 System.*out*.println("Grade: " + student.getGrade());  
 System.*out*.println("Courses: " + student.getCourses());  
 }  
}

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

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 PersonAndEmployeeExample {  
 public static void main(String[] args) {  
 Person person = new Person("John", "Doe");  
 Employee employee = new Employee("Jane", "Smith", 12345, "Manager");  
  
 System.*out*.println("Person:");  
 System.*out*.println("First Name: " + person.getFirstName());  
 System.*out*.println("Last Name: " + person.getLastName());  
  
 System.*out*.println();  
  
 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.Arrays;  
  
public class LongestConsecutiveSequence {  
// static method  
 public static int findLongestConsecutiveSequenceLength(int[] nums) {  
// here we are checking array is empty  
 if (nums.length == 0) {  
 return 0;  
 }  
  
 // Sort the array in ascending order  
 Arrays.*sort*(nums);  
  
 int maxLength = 1;  
 int currentLength = 1;  
  
 // Iterate over the sorted array  
 for (int i = 1; i < nums.length; i++) {  
 if (nums[i] != nums[i - 1]) {  
 // Check if the current number is consecutive  
 if (nums[i] == nums[i - 1] + 1) {  
 currentLength++;  
 } else {  
 // Reset the current length  
 currentLength = 1;  
 }  
  
 // 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 longestSequenceLength = *findLongestConsecutiveSequenceLength*(nums);  
 System.*out*.println("Length of the longest consecutive sequence: " + longestSequenceLength);  
 }  
}

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.io.\*;  
import java.util.\*;  
class AgeNotWithInRangeException extends Exception  
{  
 public String validage()  
 {  
 return ("Age is not between 15 and 21 … Please ReEnter the Age");  
 }  
}  
class NameNotValidException extends Exception  
{  
 public String validname()  
 {  
 return("Name is not Valid … Please ReEnter the Name");  
 }  
}  
class Student  
{  
 int roll,age;  
 String name,course;  
 Student()  
 {  
 roll=0;  
 name=null;  
 age=0;  
 course=null;  
 }  
 Student(int r,String n,int a,String c)  
 {  
 roll=r;  
 course=c;  
 int l,temp=0;  
 l=n.length();  
 for(int i=0;i<l;i++)  
 {  
 char ch;  
 ch=n.charAt(i);  
 if(ch<'A' || ch>'Z' && ch<'a' || ch>'z')  
 temp=1;  
 }  
 /\*———-Checking Name——————–\*/  
 try  
 {  
 if(temp==1)  
 throw new NameNotValidException();  
 else  
 name=n;  
 }  
 catch(NameNotValidException e2)  
 {  
 System.*out*.println(e2);  
 }  
 /\*———-Checking Age——————–\*/  
 try  
 {  
 if(a>=15 && a<=21)  
 age=a;  
 else  
 throw new AgeNotWithInRangeException();  
 }  
 catch(AgeNotWithInRangeException e1)  
 {  
 System.*out*.println(e1);  
 }  
 }  
 void display()  
 {  
 System.*out*.println("roll Name Age Course");  
 System.*out*.println("————————————————-");  
 System.*out*.println(roll+" "+name+" "+age+" "+course);  
 }  
}  
class StudentDemo  
{  
 public static void main(String args[])throws IOException  
 {  
 Scanner sc = new Scanner(System.*in*);  
 int r,a;  
 String n,c;  
  
 System.*out*.println("Enter roll,name,age,course");  
 r=sc.nextInt();  
 sc.nextLine();  
 n=sc.nextLine();  
 a=sc.nextInt();  
 sc.nextLine();  
 c=sc.nextLine();  
 Student s=new Student(r,n,a,c);  
 s.display();  
 }  
}

25. 

import java.awt.Color;  
  
import javax.swing.\*;  
public class color\_demo{  
 public static void main(String args[])  
 {  
 JFrame f = new JFrame();  
 f.setSize(550,450);  
 f.getContentPane().setBackground(Color.*GRAY*);  
 JButton b1 = new JButton("All The Best");  
 b1.setBounds(20, 10, 120, 50);  
 b1.setForeground(Color.*RED*);  
 f.add(b1);  
 JButton b2 = new JButton("All The Best");  
 b2.setBounds(145, 10, 120, 50);  
 b2.setForeground(Color.*PINK*);  
 f.add(b2);  
 JButton b3 = new JButton("All The Best");  
 b3.setBounds(270, 10, 120, 50);  
 b3.setForeground(Color.*BLUE*);  
 f.add(b3);  
 JButton b4 = new JButton("All The Best");  
 b4.setBounds(395, 10, 120, 50);  
 b4.setForeground(Color.*GREEN*);  
 f.add(b4);  
 JButton b5 = new JButton("All The Best");  
 b5.setBounds(200, 65, 120, 50);  
 f.add(b5);  
 b5.setForeground(Color.*CYAN*);  
  
 JLabel l1 = new JLabel("All The Best");  
 l1.setBounds(180, 120, 120, 50);  
 f.add(l1);  
 l1.setForeground(Color.*PINK*);;  
 JLabel l2 = new JLabel("All The Best");  
 l2.setBounds(230, 200, 120, 50);  
 f.add(l2);  
 l2.setForeground(Color.*CYAN*);  
 JLabel l3 = new JLabel("All The Best");  
 l3.setBounds(320, 280, 120, 50);  
 l3.setForeground(Color.*RED*);  
 f.add(l3);  
 JLabel l4 = new JLabel("All The Best");  
 l4.setBounds(400, 360, 120, 50);  
 l4.setForeground(Color.*black*);  
 f.add(l4);  
 f.setLayout(null);  
 f.setVisible(true);  
 }  
}