1.

1.1) Program to remove all repeated elements from an array

import java.util.ArrayList;  
  
public class ques1a {  
 public static void main(String[] args) {  
 int [] arr = new int []{10,20,10,30,20,50,30};  
 int[] uniqueArray = *removeDuplicates*(arr);  
  
 System.*out*.print("Array with repeated elements: ");  
 *printArray*(arr);  
  
 System.*out*.print("Array without repeated elements: ");  
 *printArray*(uniqueArray);  
 }  
  
 public static int[] removeDuplicates(int[] array) {  
  
 ArrayList<Integer> uniqueList = new ArrayList<>();  
  
 for (int num : array) {  
 if (!uniqueList.contains(num)) {  
 uniqueList.add(num);  
 }  
 }  
 int[] uniqueArray = new int[uniqueList.size()];  
 for (int i = 0; i < uniqueList.size(); i++) {  
 uniqueArray[i] = uniqueList.get(i);  
 }  
  
 return uniqueArray;  
 }  
  
 public static void printArray(int[] array) {  
 for (int num : array) {  
 System.*out*.print(num + " ");  
 }  
 System.*out*.println();  
 }  
}

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

import java.util.ArrayList;  
import java.util.HashSet;  
  
public class ques1b{  
 public static void main(String[] args) {  
 int[] array1 = {1, 2, 3, 4, 5};  
 int[] array2 = {4, 5, 6, 7, 8};  
 int[] commonElements = *findCommonElements*(array1, array2);  
  
 System.*out*.print("Array 1: ");  
 *printArray*(array1);  
  
 System.*out*.print("Array 2: ");  
 *printArray*(array2);  
  
 System.*out*.print("Common Elements: ");  
 *printArray*(commonElements);  
 }  
  
 public static int[] findCommonElements(int[] array1, int[] array2) {  
 HashSet<Integer> set = new HashSet<>();  
 ArrayList<Integer> commonList = new ArrayList<>();  
  
 for (int num : array1) {  
 set.add(num);  
 }  
  
 for (int num : array2) {  
 if (set.contains(num)) {  
 commonList.add(num);  
 }  
 }  
  
 int[] commonArray = new int[commonList.size()];  
 for (int i = 0; i < commonList.size(); i++) {  
 commonArray[i] = commonList.get(i);  
 }  
  
 return commonArray;  
 }  
  
 public static void printArray(int[] array) {  
 for (int num : array) {  
 System.*out*.print(num + " ");  
 }  
 System.*out*.println();  
 }  
}

2.

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

public class ques2a {  
 public static void main(String[] args) {  
  
  
 String string = "Big black bug bit a big black dog on his big black nose";  
 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 a 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 visited word  
 words[j] = "0";  
 }  
 }  
  
 //Displays the duplicate word if count is greater than 1  
 if(count > 1 && words[i] != "0")  
 System.*out*.println(words[i] +" "+ count);  
  
 }  
 }  
}

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

public class ques2b {  
 public static void main(String[] args) {  
 String str1 = "Umbrella";  
 System.*out*.println(str1.contains("e"));  
 }  
}

3.

3.1)Java Program to Reverse a String.

public class ques3a {  
 public static void main(String[] args) {  
 String str1 = "Bhargav";  
 StringBuffer sb =new StringBuffer(str1);  
 sb.reverse();  
 System.*out*.println(sb);  
  
//Another method  
 String str= "Geeks", nstr="";  
 char ch;  
  
 System.*out*.print("Original word: ");  
 System.*out*.println("Geeks"); //Example word  
  
 for (int i=0; i<str.length(); i++)  
 {  
 ch= str.charAt(i); //extracts each character  
 nstr= ch+nstr; //adds each character in front of the existing string  
 }  
 System.*out*.println("Reversed word: "+ nstr);  
 }  
}

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

public class ques3b {  
 public static void main(String[] args) {  
 String str1 = "madam";  
 StringBuffer sb = new StringBuffer(str1);  
 sb.reverse();  
 String sb1 = sb.toString();  
 if(str1.equalsIgnoreCase(sb1)){  
 System.*out*.println("Palindrome");  
 }else{  
 System.*out*.println("Non pallindrome");  
 }  
  
 }  
}

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{  
 public abstract void move();  
}  
  
class Helicopter extends Vehicle{  
 public void move(){  
 System.*out*.println("Helicopter Flies in Air");  
 }  
}  
  
class Train extends Vehicle{  
 public void move(){  
 System.*out*.println("Train runs on tracks");  
 }  
}  
  
class Car extends Vehicle{  
 public void move(){  
 System.*out*.println("Car drives on road");  
 }  
}  
public class ques4{  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter the type of vehicle");  
 String vehicleType = sc.nextLine().toLowerCase();  
 Vehicle vehicle= *createVehicle*(vehicleType);  
 if(vehicle != null){  
 vehicle.move();  
 }  
 else{  
 System.*out*.println("Invalid vehicle type");  
 }  
  
 }  
 public static Vehicle createVehicle(String vehicleType){  
 if(vehicleType.equals("helicopter")){  
 return new Helicopter();  
 } else if (vehicleType.equals("train")) {  
 return new Train();  
 }  
 else if(vehicleType.equals("car")){  
 return new Car();  
 }  
 else  
 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.

import java.util.Scanner;  
  
abstract class Marks{  
 abstract void getPercentage();  
}  
  
class A extends Marks{  
 int a1,a2,a3;  
 A(int a1,int a2,int a3){  
 this.a1=a1;  
 this.a2=a2;  
 this.a3=a3;  
 }  
 public void getPercentage(){  
 int sum= a1+a2+a3;  
 float percent = (sum\*100)/300;  
 System.*out*.print("The percentage of A is ");  
 System.*out*.print(percent);  
 }  
}  
  
class B extends Marks{  
 int b1,b2,b3,b4;  
 B(int b1,int b2,int b3, int b4){  
 this.b1=b1;  
 this.b2=b2;  
 this.b3=b3;  
 this.b4=b4;  
 }  
 public void getPercentage(){  
 int sum = b1+b2+b3+b4;  
 float percent = (sum\*100)/400;  
 System.*out*.print("The percentage of B is ");  
 System.*out*.print(percent);  
 }  
}  
public class ques5 {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("enter marks of A");  
 int a1 = sc.nextInt();  
 int a2 = sc.nextInt();  
 int a3 = sc.nextInt();  
 System.*out*.println("enter marks of B");  
 int b1 = sc.nextInt();  
 int b2 = sc.nextInt();  
 int b3 = sc.nextInt();  
 int b4 = sc.nextInt();  
  
 A a = new A(a1,a2,a3);  
 B b = new B(b1,b2,b3,b4);  
  
 a.getPercentage();  
 System.*out*.println();  
 b.getPercentage();  
 }  
}

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

**class Arithmetic:**

**def add(self, a, b):**

**return a + b**

**class Adder(Arithmetic):**

**pass**

**class Tester:**

**def main(self):**

**adder = Adder()**

**print("My superclass is:", adder.\_\_class\_\_.\_\_bases\_\_[0].\_\_name\_\_)**

**a = 42**

**b = 13**

**print(a, b, adder.add(a, b))**

**# Instantiate Tester and run the main method**

**tester = Tester()**

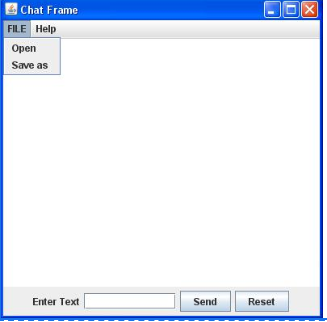
**tester.main()**

**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;  
  
  
 public 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{  
 long result = 1;  
 for (int i = 0; i < p; i++) {  
 result \*= n;  
 }  
 return result;  
 }  
  
 }  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter the number and its power");  
 int n = sc.nextInt();  
 int p = sc.nextInt();  
  
 MyCalculator obj = new MyCalculator();  
 try {  
 System.*out*.println("The answer is : " + obj.power(n, p));  
 }  
 catch(Exception e){  
 System.*out*.println("Exception occured: "+ 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.\*;  
public class PhoneBook {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter no of entries");  
 int entries = sc.nextInt();  
 Map<String,String> mp = new HashMap<>();  
 mp.put("Sejal","98867098");  
 mp.put("Badak","45678909");  
 for (int i = 0; i < entries; i++) {  
 System.*out*.println("Enter name : ");  
 String name = sc.nextLine();  
 System.*out*.println("Enter number : ");  
 String Phonenumber = sc.nextLine();  
 mp.put(name,Phonenumber);  
  
 }  
 while (sc.hasNext()) {  
 String query = sc.nextLine();  
 String phoneNumber = mp.get(query);  
  
 if (phoneNumber != null) {  
 System.*out*.println(query + " " + phoneNumber);  
 } else {  
 System.*out*.println("Not found");  
 }  
 }  
  
 sc.close();  
 }  
}

9. 

10.   
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class Adder extends JFrame implements ActionListener {  
 private JTextField firstNumberTextField;  
 private JTextField secondNumberTextField;  
 private JTextField resultTextField;  
 private JButton addButton;  
 private JButton clearButton;  
  
 public Adder() {  
 setTitle("Calculator");  
 setSize(300, 200);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setResizable(false);  
 setLayout(new GridLayout(4, 2));  
  
 JLabel firstNumberLabel = new JLabel("First Number:");  
 firstNumberTextField = new JTextField();  
 JLabel secondNumberLabel = new JLabel("Second Number:");  
 secondNumberTextField = new JTextField();  
 JLabel resultLabel = new JLabel("Result:");  
 resultTextField = new JTextField();  
 resultTextField.setEditable(false);  
  
 addButton = new JButton("Add");  
 addButton.addActionListener(this);  
 clearButton = new JButton("Clear");  
 clearButton.addActionListener(this);  
  
 add(firstNumberLabel);  
 add(firstNumberTextField);  
 add(secondNumberLabel);  
 add(secondNumberTextField);  
 add(resultLabel);  
 add(resultTextField);  
 add(addButton);  
 add(clearButton);  
  
 setVisible(true);  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == addButton) {  
 try {  
 double firstNumber = Double.*parseDouble*(firstNumberTextField.getText());  
 double secondNumber = Double.*parseDouble*(secondNumberTextField.getText());  
 double result = firstNumber + secondNumber;  
 resultTextField.setText(String.*valueOf*(result));  
 } catch (NumberFormatException ex) {  
 JOptionPane.*showMessageDialog*(this, "Invalid input! Please enter valid numbers.");  
 }  
 }  
  
 if (e.getSource() == clearButton) {  
 firstNumberTextField.setText("");  
 secondNumberTextField.setText("");  
 resultTextField.setText("");  
 }  
 }  
  
 public static void main(String[] args) {  
 SwingUtilities.*invokeLater*(new Runnable() {  
 public void run() {  
 new Adder();  
 }  
 });  
 }  
}

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 ques11a {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter the number whose table is to be printed:");  
 int n = sc.nextInt();  
  
 ques11a obj = new ques11a();  
 obj.Table(n);  
  
 }  
 public void Table(int n){  
 int m =10;  
 for(int i=1;i<=m;i++){  
 System.*out*.println(n+" x "+i+ " = "+ n\*i);  
 }  
 }  
}

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

import java.util.Scanner;  
  
public class ques11b {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter the number");  
 int n = sc.nextInt();  
 boolean flag = false;  
 if(n==1){  
 System.*out*.println("1 is neither prime nor composite number");  
 }  
 else{  
 for(int i =2;i<= n/2; i++){  
 if(n % i == 0){  
 flag = true;  
 break;  
 }  
 }  
 if(!flag){  
 System.*out*.println(n + " is a prime number");  
 }  
 else{  
 System.*out*.println(n + " is not prime number");  
 }  
 }  
  
  
  
 }  
}

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

import java.util.Scanner;  
  
public class ques12 {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.print("Enter the number of rows (half of the diamond): ");  
 int rows = sc.nextInt();  
  
  
 for (int i = 1; i <= rows; i++) {  
  
 for (int k = 1; k <= 2 \* i - 1; k++) {  
 System.*out*.print("\*");  
 }  
 System.*out*.println();  
 }  
  
  
 for (int i = rows - 1; i >= 1; i--) {  
  
 for (int k = 1; k <= 2 \* i - 1; k++) {  
 System.*out*.print("\*");  
 }  
 System.*out*.println();  
 }  
 }  
}

13.

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

import java.util.Scanner;  
  
public class ques13a {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter size of matrix");  
 int size = sc.nextInt();  
 System.*out*.println("Enter elements");  
 int [][] arr = new int[size][size];  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
  
 arr[i][j]=sc.nextInt();  
  
 }  
  
 }  
 //given matrix  
 System.*out*.println("User matrix");  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 System.*out*.print(arr[i][j]+" ");  
  
 }  
 System.*out*.println("");  
  
 }  
 System.*out*.println("Transpose of given matrix");  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 // arr[i][j]= arr[j][i];  
 System.*out*.print(arr[j][i]+" ");  
 }  
 System.*out*.println("");  
 }  
 }  
 }

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

import java.io.\*;  
public class ques13b {  
 public static void main(String[] args) throws IOException {  
 FileReader fr = new FileReader("myfile.txt");  
 BufferedReader bf =new BufferedReader(fr);  
 int wordcount = 0;  
 String line;  
 while((line=bf.readLine())!=null){  
 String[] word = line.split(" ");  
 for (String w:  
 word) {  
 wordcount++;  
  
 }  
  
 }  
 bf.close();  
 System.*out*.println("no of words in file are : "+wordcount);  
 }  
}

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[] functionButtons;  
 private JButton addButton, subtractButton, multiplyButton, divideButton;  
 private JButton equalsButton, clearButton;  
 private JPanel panel;  
  
 private double firstNumber, secondNumber, result;  
 private String operator;  
  
 public Calculator() {  
 setTitle("Basic Calculator");  
 setSize(300, 400);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setResizable(false);  
  
 panel = new JPanel();  
 panel.setLayout(new GridLayout(4, 4));  
  
 textField = new JTextField();  
 textField.setFont(new Font("Arial", Font.*PLAIN*, 18));  
 textField.setHorizontalAlignment(JTextField.*RIGHT*);  
 textField.setEditable(false);  
  
 numberButtons = new JButton[10];  
 for (int i = 0; i < 10; i++) {  
 numberButtons[i] = new JButton(String.*valueOf*(i));  
 numberButtons[i].addActionListener(this);  
 }  
  
 functionButtons = new JButton[6];  
 addButton = new JButton("+");  
 subtractButton = new JButton("-");  
 multiplyButton = new JButton("\*");  
 divideButton = new JButton("/");  
 equalsButton = new JButton("=");  
 clearButton = new JButton("C");  
  
 functionButtons[0] = addButton;  
 functionButtons[1] = subtractButton;  
 functionButtons[2] = multiplyButton;  
 functionButtons[3] = divideButton;  
 functionButtons[4] = equalsButton;  
 functionButtons[5] = clearButton;  
  
 for (int i = 0; i < 6; i++) {  
 functionButtons[i].addActionListener(this);  
 functionButtons[i].setFont(new Font("Arial", Font.*BOLD*, 16));  
 }  
  
 panel.add(numberButtons[1]);  
 panel.add(numberButtons[2]);  
 panel.add(numberButtons[3]);  
 panel.add(addButton);  
  
 panel.add(numberButtons[4]);  
 panel.add(numberButtons[5]);  
 panel.add(numberButtons[6]);  
 panel.add(subtractButton);  
  
 panel.add(numberButtons[7]);  
 panel.add(numberButtons[8]);  
 panel.add(numberButtons[9]);  
 panel.add(multiplyButton);  
  
 panel.add(numberButtons[0]);  
 panel.add(clearButton);  
 panel.add(divideButton);  
 panel.add(equalsButton);  
  
 setLayout(new BorderLayout());  
 add(textField, BorderLayout.*NORTH*);  
 add(panel, BorderLayout.*CENTER*);  
  
 setVisible(true);  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 for (int i = 0; i < 10; i++) {  
 if (e.getSource() == numberButtons[i]) {  
 textField.setText(textField.getText().concat(String.*valueOf*(i)));  
 }  
 }  
  
 if (e.getSource() == addButton) {  
 firstNumber = Double.*parseDouble*(textField.getText());  
 operator = "+";  
 textField.setText("");  
 }  
  
 if (e.getSource() == subtractButton) {  
 firstNumber = Double.*parseDouble*(textField.getText());  
 operator = "-";  
 textField.setText("");  
 }  
  
 if (e.getSource() == multiplyButton) {  
 firstNumber = Double.*parseDouble*(textField.getText());  
 operator = "\*";  
 textField.setText("");  
 }  
  
 if (e.getSource() == divideButton) {  
 firstNumber = Double.*parseDouble*(textField.getText());  
 operator = "/";  
 textField.setText("");  
 }  
  
 if (e.getSource() == equalsButton) {  
 secondNumber = Double.*parseDouble*(textField.getText());  
  
 switch (operator) {  
 case "+":  
 result = firstNumber + secondNumber;  
 break;  
 case "-":  
 result = firstNumber - secondNumber;  
 break;  
 case "\*":  
 result = firstNumber \* secondNumber;  
 break;  
 case "/":  
 if (secondNumber != 0) {  
 result = firstNumber / secondNumber;  
 } else {  
 JOptionPane.*showMessageDialog*(this, "Error: Division by zero is not allowed.");  
 }  
 break;  
 }  
  
 textField.setText(String.*valueOf*(result));  
 firstNumber = result;  
 }  
  
 if (e.getSource() == clearButton) {  
 textField.setText("");  
 }  
 }  
  
 public static void main(String[] args) {  
 SwingUtilities.*invokeLater*(new Runnable() {  
 public void run() {  
 new Calculator();  
 }  
 });  
 }  
}

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 ques15 {  
 public static void main(String[] args) {  
 ArrayList<Integer> list1 = new ArrayList<>();  
 list1.add(20);  
 list1.add(30);  
 list1.add(40);  
 System.*out*.println("Using simple for loop");  
 for (int i = 0; i < list1.size(); i++) {  
 System.*out*.print(list1.get(i)+" ");  
  
 }  
 System.*out*.println("");  
 System.*out*.println("Using advanced for loop");  
 for (Integer i:  
 list1) {  
 System.*out*.print(i+" ");  
  
 }  
 System.*out*.println("");  
 System.*out*.println("Using iterator");  
 Iterator<Integer> it = list1.iterator();  
 while (it.hasNext()){  
 System.*out*.print(it.next()+ " ");  
 }  
 }  
}

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.\*;  
public class ques16{  
 public static void main(String[] args) {  
 Map<String,Integer> l = new HashMap<>();  
 String s ="This this is is done by Saket Saket";  
 String[] sp = s.split(" ");  
 for (String words:  
 sp) {  
 if(!l.containsKey(words)){  
 l.put(words,1);  
  
 }else {  
 l.put(words,l.get(words)+1);  
 }  
  
 }  
 System.*out*.println(l);  
  
 }  
}

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 ques17 {  
 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();  
 }  
  
 Arrays.*sort*(strings);  
  
 System.*out*.println("\nSorted strings:");  
 for (String str : strings) {  
 System.*out*.println(str);  
 }  
  
 System.*out*.println("\nCombined and reversed strings:");  
 for (int i = 0; i < strings.length - 1; i += 2) {  
 String combined = strings[i] + strings[i + 1];  
 String reversed = *reverseString*(strings[i]);  
 System.*out*.println(combined + " (Reversed First: " + reversed + ")");  
 }  
  
 scanner.close();  
 }  
  
 public static String reverseString(String str) {  
 StringBuilder sb = new StringBuilder(str);  
 return sb.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{  
 private String name;  
 private int age;  
 Person(String name, int age){  
 this.name=name;  
 this.age=age;  
 }  
  
 String getName(){  
 return name;  
 }  
  
 int getAge(){  
 return age;  
 }  
}  
  
class Employee extends Person{  
 private String designation;  
 private int salary;  
  
 public Employee(String name, int age, String designation, int salary) {  
 super(name, age);  
 this.designation = designation;  
 this.salary = salary;  
 }  
 public String getDesignation(){  
 return designation;  
 }  
  
 public int getSalary(){  
 return salary;  
 }  
}  
public class ques18{  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 Employee [] emp = new Employee[5];  
 for(int i=0;i<5;i++){  
 System.*out*.println("Enter name: ");  
 String name= sc.nextLine();  
 System.*out*.println("Enter age: ");  
 int age = sc.nextInt();  
 sc.nextLine();  
 System.*out*.println("Enter Designation: ");  
 String designation= sc.nextLine();  
 System.*out*.println("Enter salary: ");  
 int salary = sc.nextInt();  
 sc.nextLine();  
 emp[i] = new Employee(name, age, designation, salary);  
 }  
  
 for (Employee employees:emp) {  
 if (employees.getSalary() > 5000) {  
 System.*out*.println(employees.getName());  
 }  
 }  
 }  
}

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.\*;  
// Define the Account interface  
interface Account {  
 void set(String name, int accountNumber, double balance);  
 void display();  
}  
  
// Define the Person interface  
interface Person1 {  
 void store(String name);  
 void disp();  
}  
  
// Implement the Customer class that extends Person and implements Account  
class Customer implements Person1, Account {  
 private String name;  
 private int accountNumber;  
 private double balance;  
  
 @Override  
 public void set(String name, int accountNumber, double balance) {  
 this.name = name;  
 this.accountNumber = accountNumber;  
 this.balance = balance;  
 }  
  
 @Override  
 public void display() {  
 System.*out*.println("Account Information:");  
 System.*out*.println("Name: " + name);  
 System.*out*.println("Account Number: " + accountNumber);  
 System.*out*.println("Balance: $" + balance);  
 }  
  
 @Override  
 public void store(String name) {  
 this.name = name;  
 }  
  
 @Override  
 public void disp() {  
 System.*out*.println("Person Information:");  
 System.*out*.println("Name: " + name);  
 }  
  
 // Calculate and display interest  
 public void calculateInterest(double interestRate) {  
 double interest = balance \* interestRate;  
 System.*out*.println("Interest: $" + interest);  
 }  
}  
  
// Main class  
public class ques19 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 // Get user input  
 System.*out*.print("Enter name: ");  
 String name = scanner.nextLine();  
  
 System.*out*.print("Enter account number: ");  
 int accountNumber = scanner.nextInt();  
  
 System.*out*.print("Enter balance: ");  
 double balance = scanner.nextDouble();  
  
 System.*out*.print("Enter interest rate: ");  
 double interestRate = scanner.nextDouble();  
  
 // Create Customer object and set values  
 Customer customer = new Customer();  
 customer.store(name);  
 customer.set(name, accountNumber, balance);  
  
 // Display information and calculate interest  
 customer.disp();  
 customer.display();  
 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{  
 String title;  
 int length;  
 boolean available;  
  
 public void show(String title, int length, boolean available){  
 this.title=title;  
 this.length=length;  
 this.available=available;  
  
 System.*out*.println("Title: "+title);  
 System.*out*.println("Length: "+length);  
 System.*out*.println("Available: "+available);  
 }  
}  
class Movie extends VideoTape{  
 float dr;  
 public void show(float dr) {  
 this.dr=dr;  
 System.*out*.println("Director rating: "+ dr);  
 }  
  
 @Override  
 public void show(String title, int length, boolean available) {  
 super.show(title, length, available);  
 }  
}  
  
class MusicVideo extends VideoTape{  
 String ac;  
 public void show(String ac) {  
 this.ac=ac;  
 System.*out*.println("Artist Category: "+ac);  
 }  
  
 @Override  
 public void show(String title, int length, boolean available) {  
 super.show(title, length, available);  
 }  
}  
public class ques20 {  
 public static void main(String[] args) {  
 Movie mv= new Movie();  
 MusicVideo mve = new MusicVideo();  
 mv.show(4.5f);  
 mve.show("pop");  
 mv.show("xyz",20,true);  
  
 }  
}

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 void displayInfo() {  
 System.*out*.println("Name: " + name);  
 System.*out*.println("Grade: " + grade);  
 }  
}  
  
public class ques21 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 // Get student's name  
 System.*out*.print("Enter student's name: ");  
 String name = scanner.nextLine();  
  
 // Get student's grade  
 System.*out*.print("Enter student's grade: ");  
 int grade = scanner.nextInt();  
  
 // Create a new student object  
 Student student = new Student(name, grade);  
  
 // Add courses  
 System.*out*.println("Enter courses (press 'q' to quit):");  
 scanner.nextLine(); // Consume the newline character  
 String course;  
 do {  
 course = scanner.nextLine();  
 if (!course.equalsIgnoreCase("q")) {  
 student.addCourse(course);  
 }  
 } while (!course.equalsIgnoreCase("q"));  
  
 // Display student information and courses  
 System.*out*.println();  
 student.displayInfo();  
 student.displayCourses();  
  
 // Remove a course  
 System.*out*.print("Enter a course to remove: ");  
 String courseToRemove = scanner.nextLine();  
 student.removeCourse(courseToRemove);  
  
 // Display updated courses  
 System.*out*.println();  
 student.displayCourses();  
  
 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 Person2 {  
 private String firstName;  
 private String lastName;  
  
 public Person2(String firstName, String lastName) {  
 this.firstName = firstName;  
 this.lastName = lastName;  
 }  
  
 public String getFirstName() {  
 return firstName;  
 }  
  
 public String getLastName() {  
 return lastName;  
 }  
}  
  
class Employee1 extends Person2 {  
 private String employeeId;  
 private String jobTitle;  
  
 public Employee1(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 super.getLastName() + " (" + jobTitle + ")";  
 }  
}  
  
public class ques22 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 // Get person's information  
 System.*out*.print("Enter person's first name: ");  
 String personFirstName = scanner.nextLine();  
  
 System.*out*.print("Enter person's last name: ");  
 String personLastName = scanner.nextLine();  
  
 // Create Person object  
 Person2 person = new Person2(personFirstName, personLastName);  
  
 // Get employee's information  
 System.*out*.print("Enter employee's first name: ");  
 String employeeFirstName = scanner.nextLine();  
  
 System.*out*.print("Enter employee's last name: ");  
 String employeeLastName = scanner.nextLine();  
  
 System.*out*.print("Enter employee's ID: ");  
 String employeeId = scanner.nextLine();  
  
 System.*out*.print("Enter employee's job title: ");  
 String jobTitle = scanner.nextLine();  
  
 // Create Employee object  
 Employee1 employee = new Employee1(employeeFirstName, employeeLastName, employeeId, jobTitle);  
  
 // Display person's information  
 System.*out*.println("\nPerson:");  
 System.*out*.println("First Name: " + person.getFirstName());  
 System.*out*.println("Last Name: " + person.getLastName());  
  
 // Display employee's information  
 System.*out*.println("\nEmployee:");  
 System.*out*.println("First Name: " + employee.getFirstName());  
 System.*out*.println("Last Name: " + employee.getLastName());  
 System.*out*.println("Employee ID: " + employee.getEmployeeId());  
  
 scanner.close();  
 }  
}

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

import java.util.Arrays;  
  
public class ques23 {  
 public static int longestConsecutive(int[] nums) {  
 if (nums == null || nums.length == 0) {  
 return 0;  
 }  
  
 Arrays.*sort*(nums);  
 int longestStreak = 1;  
 int currentStreak = 1;  
  
 for (int i = 1; i < nums.length; i++) {  
 if (nums[i] != nums[i - 1]) {  
 if (nums[i] == nums[i - 1] + 1) {  
 currentStreak++;  
 } else {  
 longestStreak = Math.*max*(longestStreak, currentStreak);  
 currentStreak = 1;  
 }  
 }  
 }  
  
 return Math.*max*(longestStreak, currentStreak);  
 }  
  
 public static void main(String[] args) {  
 int[] nums = {49, 1, 3, 200, 2, 4, 70, 5};  
 int length = *longestConsecutive*(nums);  
  
 System.*out*.println("The length of the longest consecutive elements sequence 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.

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 Student1 {  
 private int rollNo;  
 private String name;  
 private int age;  
 private String course;  
  
 public Student1(int rollNo, String name, int age, String course) throws AgeNotWithinRangeException, NameNotValidException {  
 if (age < 15 || age > 21) {  
 throw new AgeNotWithinRangeException("Age is not within the valid range (15-21).");  
 }  
  
 if (!name.matches("^[a-zA-Z\\s]+$")) {  
 throw new NameNotValidException("Name contains invalid characters.");  
 }  
  
 this.rollNo = rollNo;  
 this.name = name;  
 this.age = age;  
 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 ques24 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 try {  
 System.*out*.print("Enter roll number: ");  
 int rollNo = scanner.nextInt();  
 scanner.nextLine(); // Consume newline  
  
 System.*out*.print("Enter student name: ");  
 String name = scanner.nextLine();  
  
 System.*out*.print("Enter student age: ");  
 int age = scanner.nextInt();  
 scanner.nextLine(); // Consume newline  
  
 System.*out*.print("Enter course: ");  
 String course = scanner.nextLine();  
  
 Student1 student = new Student1(rollNo, name, age, course);  
  
 System.*out*.println("\nStudent Information:");  
 System.*out*.println("Roll No: " + student.getRollNo());  
 System.*out*.println("Name: " + student.getName());  
 System.*out*.println("Age: " + student.getAge());  
 System.*out*.println("Course: " + student.getCourse());  
 } catch (AgeNotWithinRangeException e) {  
 System.*out*.println("Error: " + e.getMessage());  
 } catch (NameNotValidException e) {  
 System.*out*.println("Error: " + e.getMessage());  
 }  
  
 scanner.close();  
 }  
}

25. 