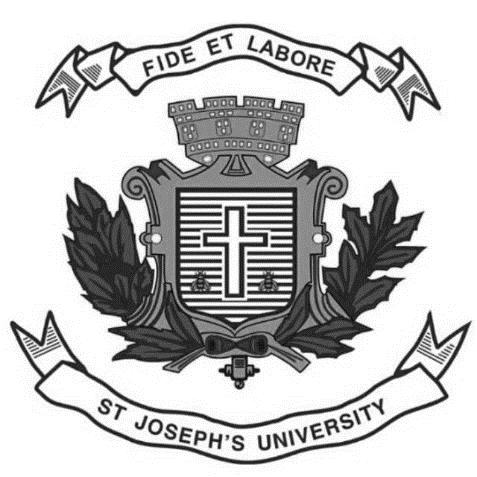
**St. Joseph’s UNIVERSITY**

**Bangalore-560027**



**Estd-1882**

**COMPUTER SCIENCE**

**PRACTICAL RECORD**

**(OBJECT ORIENTED PROGRAMMING USING JAVA)**

**Under the guidance of Submitted by:**

**1. Prof. Mary Merline Student Name: Shashank**

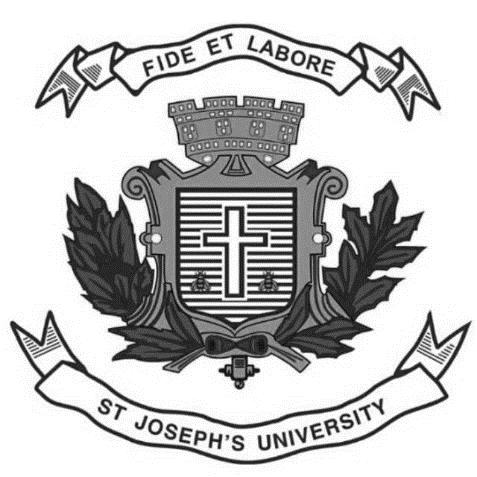
**2. Dr. Prasanthi Roll No: 222CSC33**

**3. Prof. Hubert**

**4. Prof. Selwyn**

**ST. JOSEPH’S UNIVERSITY**

**BANGALORE**



LABORATORY CERTIFICATE

This is to certify that, Sri. \_\_\_Shashank has satisfactorily completed the course of laboratory assignments in Object oriented programming using JAVA prescribed by St. Joseph’s University for the first Semester Master’s degree course in Computer Science for the year 2022-23.

Name: Shashank Teacher In-Charge

Reg. Number: 222CSC33

Date of Exam: 16/11/2022

Head of the Department/PG Coordinator

***INDEX***

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Program Names** | **Page No.** |
| 1. | Write a program to concatenate, trim and count vowels in a given string. | 01 |
| 2. | Write a program to check if a string is an anagram or a pangram. | 03 |
| 3. | Write a program to find mean, median of given numbers using command line arguments. | 05 |
| 4. | Write a program that counts the number of objects created in a class. | 07 |
| 5. | Write a program to check whether the given user password is strong or not; considering the given criteria –   1. length greater than 8, 2. At least one uppercase, 3. At least one lowercase, 4. At least one special character. | 08 |
| 6. | Write a program to implement linear search, overload this method with a multi-occurrence flag. | 10 |
| 7. | Write a program to implement a work package with a method bonus salary and call the properties of that package in an employee class. | 12 |
| 8. | Write a program to implement a work package with a method bonus 5% salary and call the properties of that package in an employee class to a set of employees in an array. | 14 |
| 9 | Write a program to implement built-in package for bubble sort. | 16 |
| 10. | Write a program to access super class in method overriding. | 18 |
| 11. | Write a program to implement multiple inheritance using interfaces. | 20 |

|  |  |  |
| --- | --- | --- |
| 12. | Write a program to implement this keyword in an inheritance class. | 22 |
| 13. | Write a program to implement try-catch block to handle:   1. Divide By Zero 2. File Not Found | 24 |
| 14. | Write a program to implement user-defined package for selection sort. | 26 |
| 15. | Write a program that implements multithreading that has 3 threads.   1. The first thread prints a random integer if the integer is even. 2. The second thread prints the square of the even number if the integer is odd. 3. The third thread prints the cube of that number. | 28 |

1. ***Write a program to concatenate, trim and count vowels in a given string***

/\*-----------------------------------------------------------

Program to perform the following string operations

1. concatenate 2 strings

2. Count the vowels in string

3. trim n elements off the string

@Author - Shashank

-------------------------------------------------------------\*/

package Printable\_java;

import java.util.Scanner;

public class StringOps {

    static String concatString(String s1, String s2) {

        return s1.concat(s2);

    }

    static int[] countVowels(String str) {

        int[] arr = { 0, 0, 0, 0, 0 };

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

            switch (str.toLowerCase().charAt(i)) {

                case 'a':

                    arr[0]++;

                    break;

                case 'e':

                    arr[1]++;

                    break;

                case 'i':

                    arr[2]++;

                    break;

                case 'o':

                    arr[3]++;

                    break;

                case 'u':

                    arr[4]++;

            }

        }

        return arr;

    }

    static String trimString(String str, int n) {

        return (str.substring(0, str.length() - n));

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("enter 2 strings");

        String s1 = sc.nextLine();

        String s2 = sc.nextLine();

        System.out.println("Concatinated string is " + concatString(s1, s2));

        System.out.println("Enter the number of elements to be trimmed from the end of the string 1");

        int n = sc.nextInt();

        System.out.println("The trimmed string 1 is " + trimString(s1, n));

        System.out.println("The counts for the vowels in String 1 are as follows");

        int[] arr = countVowels(s1);

        System.out.println("The number of a's are " + arr[0]);

        System.out.println("The number of e's are " + arr[1]);

        System.out.println("The number of i's are " + arr[2]);

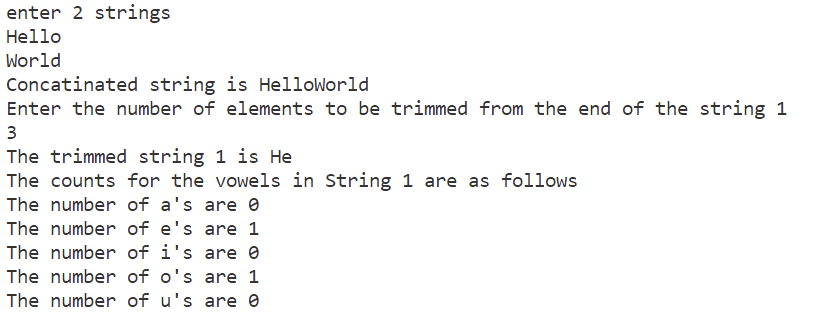
        System.out.println("The number of o's are " + arr[3]);

        System.out.println("The number of u's are " + arr[4]);

    }

}

Output:



1. ***Write a program to check if the given string is an anagram or a pangram***

/\*-----------------------------------------------------------

Program to check if a given string is anagram or pangram

@Author - Shashank

-------------------------------------------------------------\*/

package Printable\_java;

import java.util.Arrays;

import java.util.Scanner;

public class AnaPanaGram {

    static boolean PanagramCheck(String s1) {

        String alpha = "abcdefghijklmnopqrstuvwxyz";

        s1 = s1.toLowerCase();

        // checking if all letters if alpha exists in s1

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

            if (s1.indexOf(c) == -1) {

                return false;

            }

        }

        return true;

    }

    static boolean AnagramCheck(String s1, String s2) {

        s1 = s1.toLowerCase();

        s2 = s2.toLowerCase();

        // if unequal length then no need to perform any operations

        if (s1.length() != s2.length())

            return false;

        // Coverting to char array and sorting to compare

        char[] s1arr = s1.toCharArray();

        char[] s2arr = s2.toCharArray();

        Arrays.sort(s1arr);

        Arrays.sort(s2arr);

        if (Arrays.equals(s1arr, s2arr)) {

            return true;

        }

        return false;

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("enter 2 strings");

        String s1 = sc.nextLine();

        String s2 = sc.nextLine();

        if (AnagramCheck(s1, s2))

            System.out.println("The given Strings are anagrams");

        else

            System.out.println("The given strings arent anagarams");

System.out.println("Enter a string to check if panagram");

        String s3 = sc.nextLine();

        if (PanagramCheck(s3)) {

            System.out.println(s3 + " \nis a panagram");

        } else {

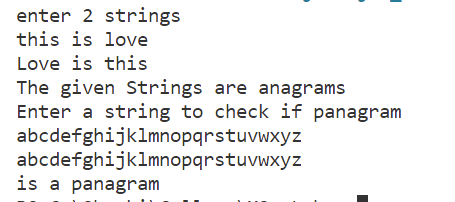
            System.out.println(s3 + " \nis a not panagram");

        }

    }

}

Output:

******

1. ***Write a program to find mean, median of given numbers using command line arguments***

/\*-----------------------------------------------------------

Program to calculate mean median and mode of command line arguments

@Author - Shashank

-------------------------------------------------------------\*/

package Printable\_java;

import java.util.Arrays;

public class StatsOpsCLargs {

    public static void main(String[] args) {

        System.out.println("The command line arguments given are");

        System.out.println(Arrays.toString(args));

        double[] array = new double[args.length];

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

            array[i] = Double.parseDouble(args[i]);

        }

        System.out.println("Given array is " + Arrays.toString(array));

        System.out.println("Mean is " + calcMean(array));

        System.out.println("Median is " + calcMedian(array));

        System.out.println("Mode is " + calcMode(array));

    }

    private static double calcMode(double[] array) {

        Arrays.sort(array);

        // using the emperical relation to calculate the mode ie

        // mode = mean - (3\*(mean-median))

        double mode = calcMean(array) - 3 \* (calcMean(array) - calcMedian(array));

        return mode;

    }

    private static double calcMedian(double[] array) {

        int size = array.length;

        Arrays.sort(array);

        if (size % 2 != 0) {

            return array[size / 2];

        }

        return (array[size / 2 - 1] + array[size / 2]) / 2;

    }

    private static double calcMean(double[] array) {

        double sum = 0;

        for (double x : array) {

            sum += x;

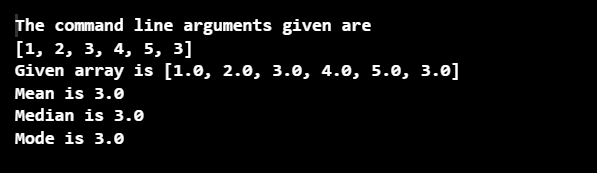
        }

        return sum / array.length;

    }

}

Output:



1. ***Write a program that counts the number of objects created in a class.***

/\*-----------------------------------------------------------

Program to count the number of instances a class contains

@Author - Shashank

-------------------------------------------------------------\*/

package Printable\_java;

class CountEmployeeClass {

    static private int obj\_count = 0;

    public static int getCount() {

        return obj\_count;

    }

    public static void plusCounter() {

        obj\_count++;

    }

}

class Employee {

    String name;

    double salary, age;

    Employee(String name, int salary, int age) {

        this.name = name;

        this.age = age;

        this.salary = salary;

    }

    // non- static block increasing the counter when an object is created

    {

        CountEmployeeClass.plusCounter();

    }

}

public class ObjCounter {

    public static void main(String[] args) throws IllegalArgumentException, IllegalAccessException {

        Employee e1 = new Employee("shashank", 10000, 25);

        Employee e2 = new Employee("shank", 100060, 21);

        Employee e3 = new Employee("shanki", 1000, 23);

        System.out.println("The number of objects in Employee class is " + CountEmployeeClass.getCount());

    }

}

Output:



1. ***Write a program to check whether the given password is strong or not based on the following***
   1. ***Length greater than 7***
   2. ***At least one lowercase letter***
   3. ***At least one uppercase letter***
   4. ***At least one digit***
   5. ***At least one special character***

/\*-----------------------------------------------------------

Program to validate passwords with the given conditions

@Author - Shashank

-------------------------------------------------------------\*/

package Printable\_java;

import java.util.Scanner;

public class PasswordValidator {

    public static void main(String[] args) {

        String password;

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter your password");

        password = sc.nextLine();

        if (passwordValidate(password)) {

            System.out.println("Your password is strong");

            return;

        }

        System.out.println("Your password is weak");

    }

    private static boolean passwordValidate(String password) {

        if (password.length() < 8) {

            return false;

        }

        // flags

        boolean ucase = false, lcase = false, special = false, num = false;

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

            Character c = password.charAt(i);

            if (Character.isLowerCase(c)) {

                lcase = true;

            } else if (Character.isUpperCase(c)) {

                ucase = true;

            } else if (Character.isDigit(c)) {

                num = true;

                // considering all characters except space and above conditions as special

                // characters

            } else if (!Character.isWhitespace(c))

                special = true;

            else

                return false;

        }

        if (lcase && ucase && special && num)

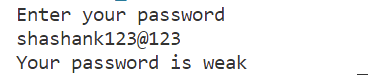
            return true;

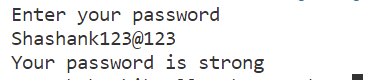
        return false;

    }

}

Outputs:

******

******

1. ***Write a program to implement linear search, overload this method with a multi-occurrence flag.***

/\*-----------------------------------------------------------

Program to implement linear search with a

multi linear search overload

@Author - Shashank

-------------------------------------------------------------\*/

package Printable\_java;

import java.util.ArrayList;

import java.util.Scanner;

public class LinSearchOverload {

    public static void linearSearch(int element, int[] array) {

        boolean flag = false;

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

            if (element == array[i]) {

                flag = true;

                System.out.println("Element found in the index " + i);

                break;

            }

        }

        if (flag == false)

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

    }

    // method overload for multi-linear search

    public static void linearSearch(int element, int[] array, boolean multiple) {

        if (multiple) {

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

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

                if (element == array[i]) {

                    newArr.add(i);

                }

            }

            if (newArr.isEmpty())

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

            else

                System.out.println("element found in");

            System.out.println(newArr.toString());

        } else {

            // calling the other overload if multi-flag is off

            linearSearch(element, array);

        }

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int n = sc.nextInt();

        int[] array = new int[n];

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

        for (int i = 0; i < n; i++) {

            array[i] = sc.nextInt();

        }

        System.out.println("Enter the element to be searched in the array");

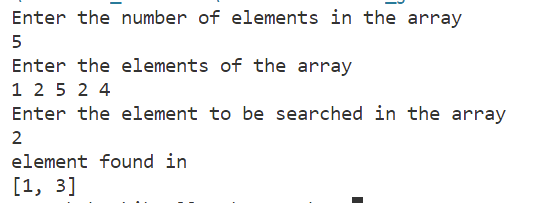
        int element = sc.nextInt();

        linearSearch(element, array, true);

    }

}

***Output:***

******

1. ***Write a program to implement a work package with a method bonus salary and call the properties of that package in an employee class.***

/\*-----------------------------------------------------------

Program to provide a list employees bonus of 5%

@Author - Shashank

-------------------------------------------------------------\*/

package Printable\_java;

import java.util.Scanner;

import Printable\_java.Work.Employee;

public class EmpBonus1 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int n = sc.nextInt();

        sc.nextLine();

        // creating an array of Employee objects

        Employee[] emp\_list = new Employee[n];

        System.out.println("Enter the name, salary and age of the list of employees");

        for (int i = 0; i < n; i++) {

            // initializing each member of the array

            System.out.println("Enter details of employee number " + (i + 1));

            String name = sc.nextLine();

            int salary = sc.nextInt();

            int age = sc.nextInt();

            sc.nextLine();

            emp\_list[i] = new Employee(name, salary, age);

        }

        // Adding bonus to salary

        emp\_list = Employee.addBonusToEmpList(emp\_list);

        // displaying

        Employee.display(emp\_list);

    }

}

Work package: Employee class

package Printable\_java.Work;

public class Employee {

    String name;

    int salary;

    int age;

    public Employee(String name, int salary, int age) {

        this.age = age;

        this.name = name;

        this.salary = salary;

    }

    public static Employee[] addBonusToEmpList(Employee[] list) {

        for (Employee e : list) {

            e.salary += e.salary \* .05;

        }

        return list;

    }

    public static void display(Employee[] list) {

        for (Employee e : list) {

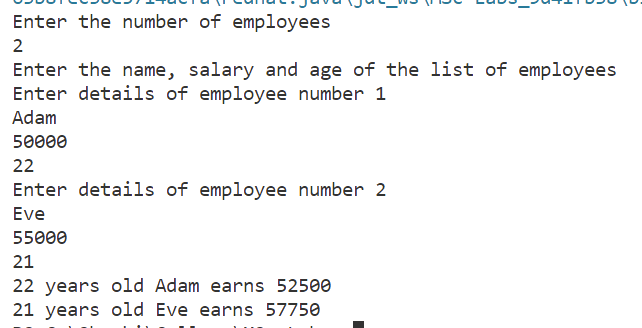
            System.out.println(e.age + " years old " + e.name + " earns " + e.salary);

        }

    }

}

Output:



1. ***Write a program to implement a work package with a method bonus 5% salary and call the properties of that package in an employee class to a set of employees in an array.***

/\*-----------------------------------------------------------

Program to provide a list employees bonus of 5%

@Author - Shashank

-------------------------------------------------------------\*/

package Printable\_java;

import java.util.Scanner;

import Printable\_java.Work.Employee;

public class EmpBonus2 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int n = sc.nextInt();

        sc.nextLine();

        // creating an array of Employee objects

        Employee[] emp\_list = new Employee[n];

        System.out.println("Enter the name, salary and age of the list of employees");

        for (int i = 0; i < n; i++) {

            // initializing each member of the array

            System.out.println("Enter details of employee number " + (i + 1));

            String name = sc.nextLine();

            int salary = sc.nextInt();

            int age = sc.nextInt();

            sc.nextLine();

            emp\_list[i] = new Employee(name, salary, age);

        }

        // Adding bonus to salary

        emp\_list = Employee.addBonusToEmpList(emp\_list);

        // displaying

        Employee.display(emp\_list);

    }

}

Work package: Employee Class

package Printable\_java.Work;

public class Employee {

    String name;

    int salary;

    int age;

    public Employee(String name, int salary, int age) {

        this.age = age;

        this.name = name;

        this.salary = salary;

    }

    public static Employee[] addBonusToEmpList(Employee[] list) {

        for (Employee e : list) {

            e.salary += e.salary \* .05;

        }

        return list;

    }

    public static void display(Employee[] list) {

        for (Employee e : list) {

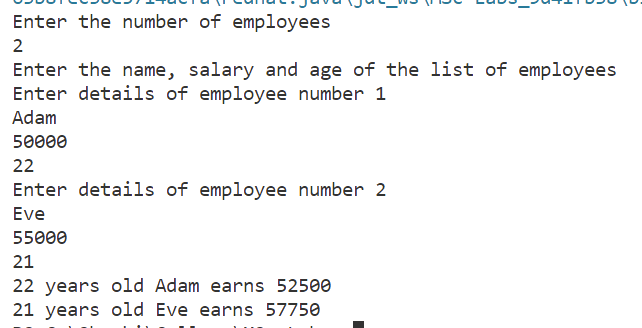
            System.out.println(e.age + " years old " + e.name + " earns " + e.salary);

        }

    }

}

Output:



1. ***Write a program to implement built-in package for bubble sort.***

/\*-----------------------------------------------------------

Program to implement bubble sort and use built in packages

@Author - Shashank

-------------------------------------------------------------\*/

package Printable\_java;

import java.util.Arrays;

import java.util.Scanner;

public class BubbleSort {

    private static int[] sortOp(int[] arr) {

        // FunctionMethod to perform Bubble Sort

        // arr[] is the input array to be sorted

        int temp = 0;

        // temp is used during a swapping operation

        boolean swapChecker = false;

        // swapChecker is used to check if elements are already swapped (sorted)

        for (int ind1 = 0; ind1 < arr.length - 1; ind1++) { // Sorting iteration

            swapChecker = false;

            for (int ind2 = 0; ind2 < arr.length - ind1 - 1; ind2++) {

                // Swapping iteration

                if (arr[ind2] > arr[ind2 + 1]) {

                    temp = arr[ind2];

                    arr[ind2] = arr[ind2 + 1];

                    arr[ind2 + 1] = temp;

                    swapChecker = true;

                    // Once swapChecker is true, next sorting iteration is

                    // performed or the sorted array is printed if all elements

                    // are sorted

                }

            }

            if (swapChecker != true) {

                // Sorting iteration will continue as long as swapChecker is false

                break;

            }

        }

        return arr; // Returns the sorted array

    }

    private static void printResult(int[] arr) {

        // Function/Method to print the sorted array

        System.out.println(Arrays.toString(arr) + "\n" +

                "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        String spacingLine = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

        System.out.println("\nEnter the size of the array to be inputted:");

        int size = scan.nextInt();

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

        int[] array = new int[size];

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

            array[ind] = scan.nextInt();

        }

        System.out.println("\nThe array you inputted is:");

        System.out.println(Arrays.toString(array));

        System.out.println("\n" + spacingLine);

        System.out.println("Performing Bubble Sort now!");

        System.out.println(spacingLine);

        System.out.println("The sorted array is:\n");

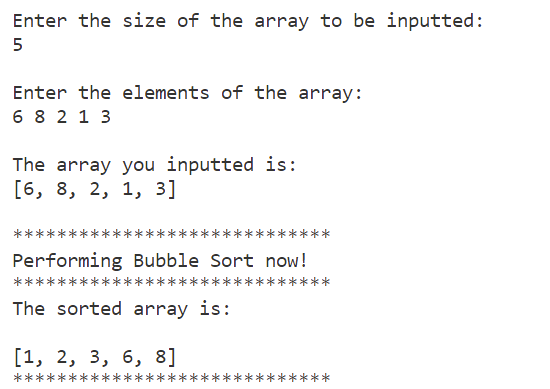
        printResult(sortOp(array));

        scan.close();

    }

}

***Output:***

******

1. ***Write a program to access super class in method overriding.***

/\*-----------------------------------------------------------

Program to demonstrate parent method calls using super keyword

@Author - Shashank

-------------------------------------------------------------\*/

package Printable\_java;

import java.util.Arrays;

import java.util.Scanner;

class Parent {

    public int binarySearch(int[] arr, int element) {

        int pos = Arrays.binarySearch(arr, element);

        return pos;

    }

}

class Child extends Parent {

    public int binarySearch(int[] arr, int element) {

        int pos = super.binarySearch(arr, element);

        System.out.println("Element found in " + pos);

        return pos;

    }

}

public class SuperOverride {

    public static void main(String[] args) {

        Child obj = new Child();

        Scanner sc = new Scanner(System.in);

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

        int n = sc.nextInt();

        int[] array = new int[n];

        System.out.println("Enter " + n + " elements");

        for (int i = 0; i < n; i++) {

            array[i] = sc.nextInt();

        }

        System.out.println("Enter the number to be searched");

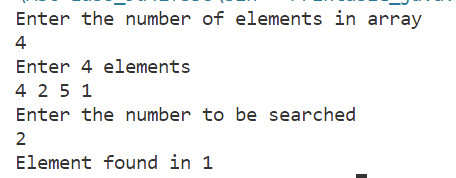
        int element = sc.nextInt();

        obj.binarySearch(array, element);

    }

}

Output:



1. ***Write a program to implement multiple inheritance using interfaces***

/\*-----------------------------------------------------------

Program to demonstrate multi level inheritence using interfaces

@Author - Shashank

-------------------------------------------------------------\*/

package Printable\_java;

interface myInterface {

    public void display(int[] arr);

}

class myClass {

    int smallest(int[] arr) {

        return -1;

    }

    int largest(int[] arr) {

        return -1;

    }

}

public class MultiInherit extends myClass implements myInterface {

    int smallest(int[] arr) {

        int min = arr[0];

        for (int x : arr) {

            if (x < min)

                min = x;

        }

        return min;

    }

    int largest(int[] arr) {

        int max = arr[0];

        for (int x : arr) {

            if (x > max)

                max = x;

        }

        return max;

    }

    public void display(int[] arr) {

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

            System.out.printf("%d ", arr[i]);

        }

        System.out.println();

    }

    public void display(int[] arr, int n) {

        for (int i = 0; i < n; i++) {

            System.out.printf("%d ", arr[i]);

        }

        System.out.println();

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        MultiInherit obj = new MultiInherit();

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

        int n = sc.nextInt();

        int[] array = new int[n];

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

        for (int i = 0; i < n; i++) {

            array[i] = sc.nextInt();

        }

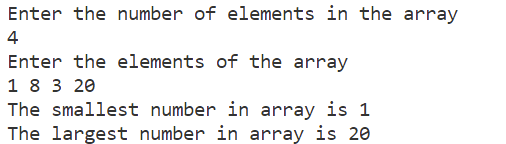
        System.out.println("The smallest number in array is " + obj.smallest(array));

        System.out.println("The largest number in array is " + obj.largest(array));

    }

}

Output:

******

1. ***Write a program to implement this keyword in an inheritance class.***

/\*-----------------------------------------------------------

Program to demonstrate the use of this keyword

and to calculate gcd, lcm of 2 numbers

@Author - Shashank

-------------------------------------------------------------\*/

package Printable\_java;

import java.util.Scanner;

public class ThisClassOuter {

    class NumericalValues {

        int num1 = 0;

        int num2 = 1;

    }

    public class ThisClass extends NumericalValues {

        private void mathOp(int num1, int num2) {

            this.num1 = num1;

            this.num2 = num2;

            int gcd = 1;

            for (int i = 1; i <= num1 && i <= num2; i++) {

                if (num1 % i == 0 && num2 % i == 0) {

                    gcd = i;

                }

            }

            int lcm = 0;

            lcm = (num1 \* num2) / gcd;

            System.out.println("Finding the GCD of " + num1 + " and " + num2 +

                    "!\n");

            System.out.println("The GCD is: " + gcd);

            System.out.println("\nFinding the LCM of " + num1 + " and " + num2 +

                    "!\n");

            System.out.println("The LCM is: " + lcm);

        }

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter 2 numbers:\n");

        int num1 = scan.nextInt();

        int num2 = scan.nextInt();

        ThisClassOuter o = new ThisClassOuter();

        // uses the outer object to create a class of inner class

        ThisClass result = o.new ThisClass();

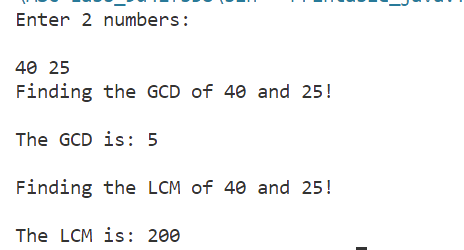
        result.mathOp(num1, num2);

        scan.close();

    }

}

Output:



1. ***Write a program to implement try catch block and handle divide by 0 and file not found exceptions***

/\*-----------------------------------------------------------

Program to Demonstate exception handling

Catching filenotfound and divide by 0 errors

@Author - Shashank

-------------------------------------------------------------\*/

package Printable\_java;

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

import java.util.Scanner;

public class ExceptionHandler {

    private static void divideOp(int num1, int num2) {

        int result = 0;

        try {

            // Placing the block of code that may throw a ArithmeticException within

            // the try block

            result = num1 / num2;

            System.out.println("The result of the division is " + result);

        } catch (ArithmeticException e1) {

            System.out.println("It is not possible to divide a number by 0!");

        }

    }

    private static void readFile(String fileName) {

        FileReader fileReader;

        try {

            // Placing the block of code that may throw a FileNotFound within

            // the try block

            fileReader = new FileReader(fileName);

            BufferedReader bufferedReader = new BufferedReader(fileReader);

            String fileData = null;

            try {

                // Placing the block of code that may throw a IOException

                // within the try block

                while ((fileData = bufferedReader.readLine()) != null) {

                    System.out.println(fileData);

                }

            } catch (IOException e2) {

                e2.printStackTrace(); // prints the message of e2 & the line

                // number where the exception occurs

            }

        } catch (FileNotFoundException e3) {

            e3.printStackTrace();

            // prints the message of e3 &

            // the line number where the exception occurs

        }

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter the name of the file to be printed");

        readFile(scan.nextLine());

        System.out.println("Enter 2 numbers:");

        int num1 = scan.nextInt();

        int num2 = scan.nextInt();

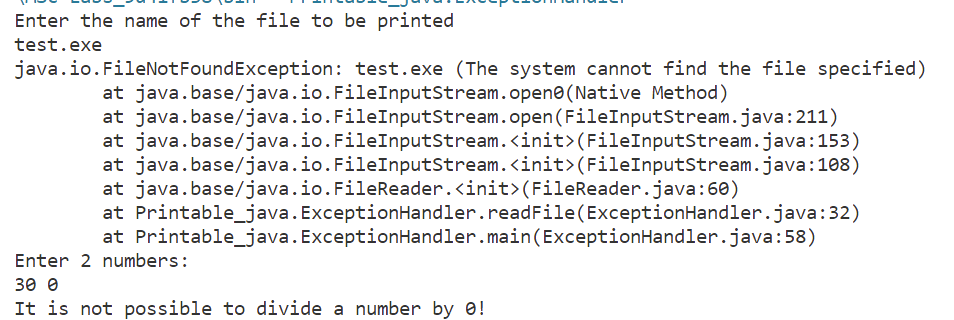
        divideOp(num1, num2);

        scan.close();

    }

}

Output:



1. ***Write a program to implement user-defined package for selection sort.***

/\*-----------------------------------------------------------

Program to Demonstarte user-defined packages

and perform selection sort

@Author - Shashank

-------------------------------------------------------------\*/

package Printable\_java;

import java.util.Arrays;

import java.util.Scanner;

import Printable\_java.Work.\*;;

public class SelectionDriver {

    public static void main(String[] args) {

        SelectionSort obj = new SelectionSort();

        Scanner scan = new Scanner(System.in);

        System.out.println("\nEnter the size of the array to be inputted:");

        int size = scan.nextInt();

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

        int[] array = new int[size];

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

            array[ind] = scan.nextInt();

        }

        System.out.println("\nThe array you inputted is:");

        System.out.println(Arrays.toString(array));

        array = obj.selectionSort(array);

        System.out.println("Sorted array is " + Arrays.toString(array));

    }

}

Work package: SelectionSort class

package Printable\_java.Work;

public class SelectionSort {

    public int[] selectionSort(int[] unsorted) {

        System.out.println("Using selection sort");

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

            int min\_index = i;

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

                if (unsorted[min\_index] > unsorted[j]) {

                    min\_index = j;

                }

            }

            int t = unsorted[i];

            unsorted[i] = unsorted[min\_index];

            unsorted[min\_index] = t;

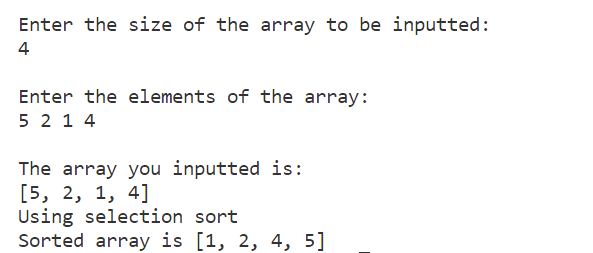
        }

        return (unsorted);

    }

}

Output:



1. ***Write a program that implements multithreading that has 3 threads***
   1. ***The first thread prints a random integer***
   2. ***The second square the integer first thread provides***
   3. ***The third thread cubes the integer first provides***

/\*-----------------------------------------------------------

Program to demonstrate multithreading

@Author - Shashank

-------------------------------------------------------------\*/

package Printable\_java;

public class MultiThreadApp extends Thread {

    public int rand;

    public class Job1 extends Thread {

        // Thread one generates the random integer

        public void run() {

            while (true) {

                try {

                    sleep(1000);

                } catch (InterruptedException e) {

                    e.printStackTrace();

                }

                // Creates a random number between 0 to 10

                rand = (int) (Math.random() \* (10 - 0 + 1) + 0);

                System.out.println("Original number " + rand);

            }

        }

    }

    // Job2 thread squares and displays the generated integer

    public class Job2 extends Thread {

        public void run() {

            while (true) {

                if (rand % 2 == 0)

                    System.out.println("Squared of original " + rand \* rand);

                try {

                    sleep(1000);

                } catch (InterruptedException e) {

                    // TODO Auto-generated catch block

                    e.printStackTrace();

                }

            }

        }

    }

    // job3 cubes and displays the generated integer

    public class Job3 extends Thread {

        public void run() {

            while (true) {

                if (rand % 2 == 1)

                    System.out.println("Cube of original " + rand \* rand \* rand);

                try {

                    sleep(1000);

                } catch (InterruptedException e) {

                    // TODO Auto-generated catch block

                    e.printStackTrace();

                }

            }

        }

    }

    public static void main(String[] args) {

        MultiThreadApp obj = new MultiThreadApp();

        obj.new Job1().start();

        obj.new Job2().start();

        obj.new Job3().start();

    }

}

Output:

