**Pandit Deendayal Energy University**

**Gandhinagar**



Object Oriented Programming in Java

Name: Dev Sapariya

Roll no.: 21BCP293

Branch: Computer Science Engineering (Division 5, Group 9)

Course Code: 20CP204P

**Module 1**

Question 1:

Write a program which will greet you with your name.

**Code**

public class Q1{

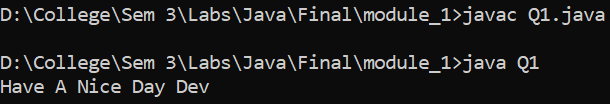
    public static void main(String[] args) {

        System.out.println("Have A Nice Day Dev");

    }

}

**Input/Output**



Question 2:

Write a program which will print greeting message for your 5 friends. Friends name should be entered as command line arguments.

**Code**

public class Q2 {

    public static void main(String[] args) {

        System.out.println("Have A Nice Day "+args[0]);

        System.out.println("Have A Nice Day "+args[1]);

        System.out.println("Have A Nice Day "+args[2]);

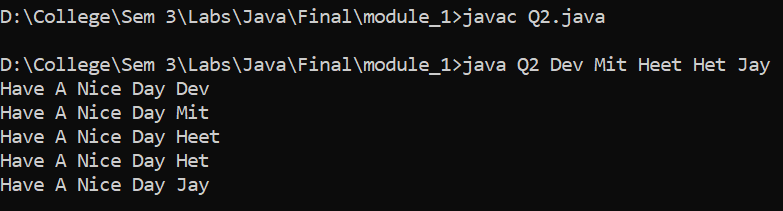
        System.out.println("Have A Nice Day "+args[3]);

        System.out.println("Have A Nice Day "+args[4]);

    }

}

**Input/Output**

****

Question 3:

Write a program which will print properties like size, min-value and max-value of each primitive number types in java.(Use appropriate wrapper class).

**Code**

public class Q3{

    public static void main(String[] args) {

        System.out.println("Size of byte: "+Byte.SIZE);

        System.out.println("Max value of byte: "+Byte.MAX\_VALUE);

        System.out.println("Min value of byte: "+Byte.MIN\_VALUE);

        System.out.println("Size of short: "+Short.SIZE);

        System.out.println("Max value of short: "+Short.MAX\_VALUE);

        System.out.println("Min value of short: "+Short.MIN\_VALUE);

        System.out.println("Size of int: "+Integer.SIZE);

        System.out.println("Max value of int: "+Integer.MAX\_VALUE);

        System.out.println("Min value of int: "+Integer.MIN\_VALUE);

        System.out.println("Size of long: "+Long.SIZE);

        System.out.println("Max value of long: "+Long.MAX\_VALUE);

        System.out.println("Min value of long: "+Long.MIN\_VALUE);

        System.out.println("Size of float: "+Float.SIZE);

        System.out.println("Max value of float: "+Float.MAX\_VALUE);

        System.out.println("Min value of float: "+Float.MIN\_VALUE);

        System.out.println("Size of double: "+Double.SIZE);

        System.out.println("Max value of double: "+Double.MAX\_VALUE);

        System.out.println("Min value of double: "+Double.MIN\_VALUE);

        System.out.println("Size of char: "+Character.SIZE);

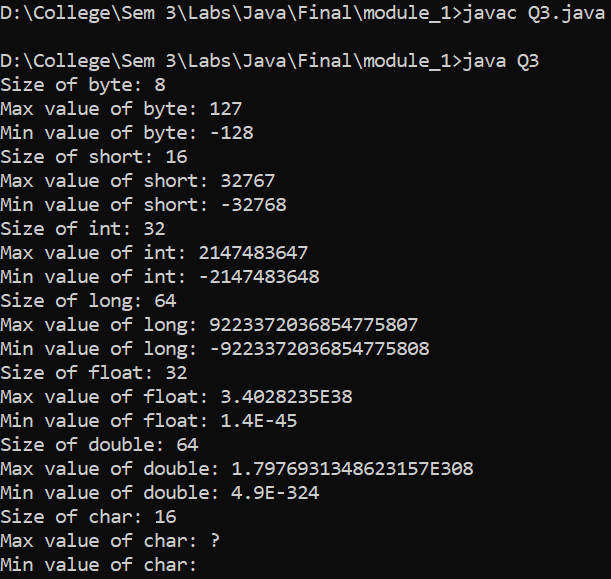
        System.out.println("Max value of char: "+Character.MAX\_VALUE);

        System.out.println("Min value of char: "+Character.MIN\_VALUE);

    }

}

**Input/Output**

****

Question 4:

Write a program which will print result of student according to marks.like A grade – marks 90 to 100, B grade – marks 80 to 90, C grade- marks 60 to 80.D grade – marks 45 to 60.E grade – marks 35 to 45. Fail – marks below 35.(Use Else-if ladder).

**Code**

public class Q4 {

    public static void main(String[] args) {

        int marks = 90;

        System.out.println("Marks: "+marks);

        if (marks >= 90 && marks <= 100) {

            System.out.println("A grade");

        } else if (marks >= 80 && marks < 90) {

            System.out.println("B grade");

        } else if (marks >= 60 && marks < 80) {

            System.out.println("C grade");

        } else if (marks >= 45 && marks < 60) {

            System.out.println("D grade");

        } else if (marks >= 35 && marks < 45) {

            System.out.println("E grade");

        } else if (marks < 35) {

            System.out.println("Fail");

        } else {

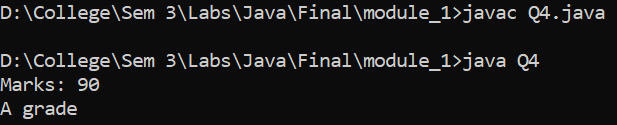
            System.out.println("Invalid marks");

        }

    }

}

**Input / Output**

****

Question 5:

Write a program which demonstrate conditional operator, compound assignment operator, pre-post increment and decrement operator, bitwise operator, logical short circuit operator.

**Code**

public class Q5 {

    public static void main(String[] args) {

        int a = 40;

        int b = 30;

        int c = 20;

        System.out.println("the value of a is " + a);

        System.out.println("the value of b is " + b);

        System.out.println("the value of c is " + c);

        if (a > b && a > c) {

            System.out.println("a is greater");

        }

        if (a > b || a > c) {

            System.out.println("a may be greater than c or b");

        }

        a += b;

        System.out.println("the value of a after a+=b is " + a);

        a++;

        System.out.println("the value of a after a++ is " + a);

        --a;

        System.out.println("the value of a after --a is " + a);

        System.out.println("a & b = " + (a & b));

        System.out.println("a | b = " + (a | b));

        System.out.println("a ^ b = " + (a ^ b));

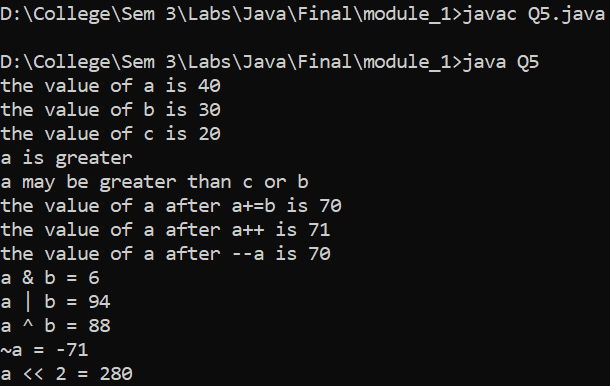
        System.out.println("~a = " + ~a);

        System.out.println("a << 2 = " + (a << 2));

    }

}

**Input / Output**

****

Question 6:

Write a program to print first N prime numbers.

**Code**

import java.util.Scanner;

public class Q6 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int n = sc.nextInt();

        int count = 0;

        int i = 2;

        while (count < n) {

            if (isPrime(i)) {

                System.out.println(i);

                count++;

            }

            i++;

        }

    }

    public static boolean isPrime(int n) {

        for (int i = 2; i < n; i++) {

            if (n % i == 0) {

                return false;

            }

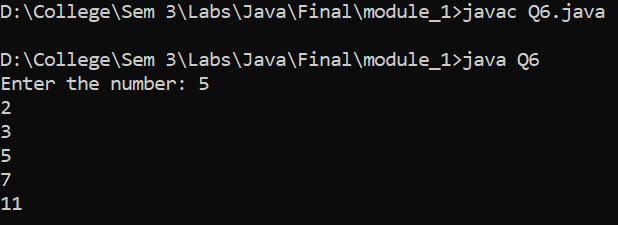
        }

        return true;

    }

}

**Input / Output**

****

Question 7:

Write a program which will create an array of integers and perform following operation:

1) Sum 2) product 3) Numbers divisible by 15 4) Maximum Value

5) Minimum Value 6) Sort

**Code**

public class Q7 {

    public static void main(String[] args) {

        int[] arr = { 15, 12, 10, 8, 5, 0, 7, 8, 9, 10 };

        int sum = 0;

        int product = 1;

        int count = 0;

        int max = arr[0];

        int min = arr[0];

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

            sum += arr[i];

            product \*= arr[i];

            if (arr[i] % 15 == 0) {

                count++;

            }

            if (arr[i] > max) {

                max = arr[i];

            }

            if (arr[i] < min) {

                min = arr[i];

            }

        }

        System.out.println("Sum: " + sum);

        System.out.println("Product: " + product);

        System.out.println("Numbers divisible by 15: " + count);

        System.out.println("Maximum Value: " + max);

        System.out.println("Minimum Value: " + min);

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

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

                if (arr[i] > arr[j]) {

                    int temp = arr[i];

                    arr[i] = arr[j];

                    arr[j] = temp;

                }

            }

        }

        System.out.print("Sorted Array: ");

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

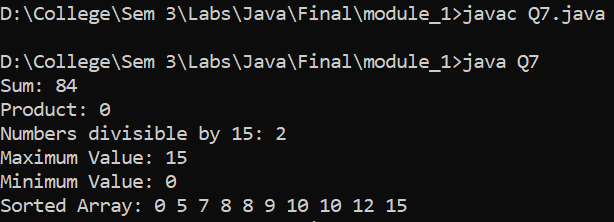
            System.out.print(arr[i] + " ");

        }

    }

}

**Input / Output**

****

Question 8:

Write an interactive program to print a string entered in a pyramid form. For instance, the string “stream” has to be displayed as follows:

S

S t

S t r

S t r e

S t r e a

S t r e a m

**Code**

import java.util.Scanner;

public class Q8 {

    public static void main(String[] args) {

        System.out.print("Enter the String: ");

        Scanner sc = new Scanner(System.in);

        String a= sc.nextLine();

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

            for (int j = 0; j < a.length() - i; j++) {

                System.out.print(" ");

            }

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

                System.out.print(a.charAt(j) + " ");

            }

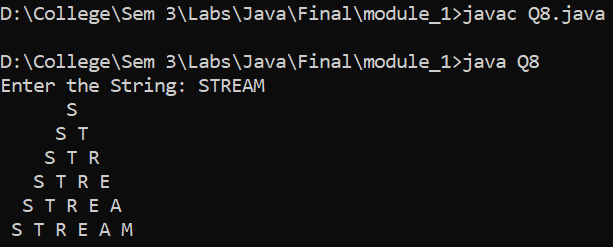
            System.out.println();

        }

    }

}

**Input / Output**

****

Question 9:

Print following diamond pattern :

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

**Code**

import java.util.Scanner;

public class Q9 {

    public static void main(String[] args) {

        System.out.print("Enter the number of Maximum number of diamond in the row: ");

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

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

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

                System.out.print(" ");

            }

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

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

            }

            System.out.println();

        }

        for (int i = 0; i < n; i++) {

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

                System.out.print(" ");

            }

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

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

            }

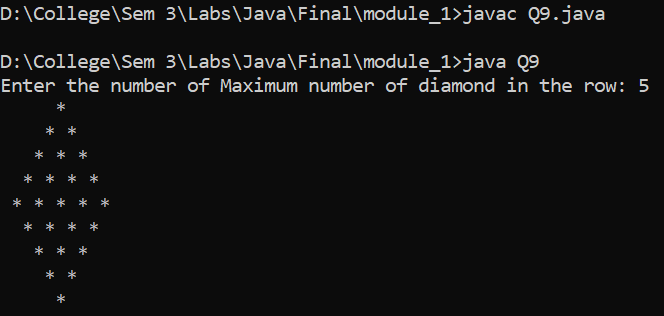
            System.out.println();

        }

    }

}

**Input / Output**



Question 10:

Write a program to accept a line and check how many consonants and vowels are there in line.

**Code**

import java.util.Scanner;

public class Q10 {

    public static void main(String[] args) {

        System.out.print("Enter the String: ");

        Scanner sc = new Scanner(System.in);

        String a = sc.nextLine();

        int v = 0;

        int c = 0;

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

            if (a.charAt(i) == 'a' || a.charAt(i) == 'e' || a.charAt(i) == 'i' || a.charAt(i) == 'o' || a.charAt(i) == 'u') {

                v++;

            }

            else if(a.charAt(i)==' '){

                continue;

            }

            else {

                c++;

            }

        }

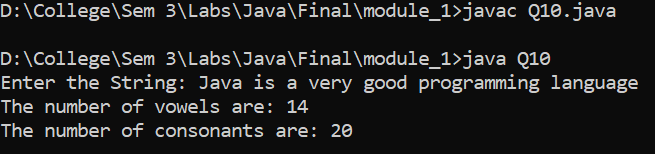
        System.out.println("The number of vowels are: " + v);

        System.out.println("The number of consonants are: " + c);

    }

}

**Input / Output**



Question 11:

Write a program to count the number of words that start with capital letters.

**Code**

import java.util.Scanner;

public class Q11 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the String: ");

        String a = sc.nextLine();

        int count = 0;

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

            if (a.charAt(i) >= 'A' && a.charAt(i) <= 'Z') {

                count++;

            }

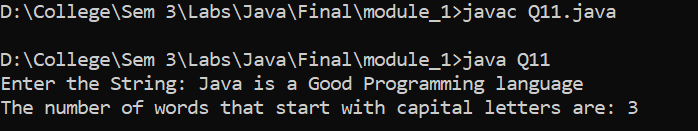
        }

        System.out.println("The number of words that start with capital letters are: " + count);

    }

}

**Input / Output**



Question 12:

Create a class which ask the user to enter a sentence, and it should display count of each vowel type in the sentence. Display the total count of each vowel and digits for all sentences.

**Code**

import java.util.Scanner;

public class Q12 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the String: ");

        String string = sc.nextLine();

        int count = 0;

        int a = 0;

        int e = 0;

        int i = 0;

        int o = 0;

        int u = 0;

        int digit = 0;

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

            if (string.charAt(j) == 'a' || string.charAt(j) == 'A') {

                a++;

            } else if (string.charAt(j) == 'e' || string.charAt(j) == 'E') {

                e++;

            } else if (string.charAt(j) == 'i' || string.charAt(j) == 'I') {

                i++;

            } else if (string.charAt(j) == 'o' || string.charAt(j) == 'O') {

                o++;

            } else if (string.charAt(j) == 'u' || string.charAt(j) == 'U') {

                u++;

            } else if (string.charAt(j) == ' ') {

                continue;

            } else if (string.charAt(j) >= '0' && string.charAt(j) <= '9') {

                digit++;

            } else {

                count++;

            }

        }

        System.out.println("The total number of a in the string is: " + a);

        System.out.println("The total number of e in the string is: " + e);

        System.out.println("The total number of i in the string is: " + i);

        System.out.println("The total number of o in the string is: " + o);

        System.out.println("The total number of u in the string is: " + u);

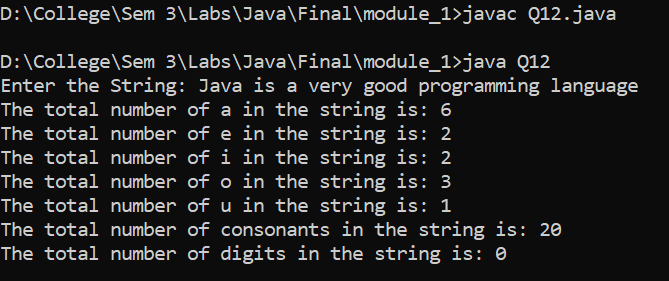
        System.out.println("The total number of consonants in the string is: " + count);

        System.out.println("The total number of digits in the string is: " + digit);

    }

}

**Input / Output**



Question 13:

Write a program which will perform following functionality on String:

a) convert to uppercase b) convert to lowercase c) count total words d) substring e) If String ends with Java then concat standardEdition to it f) trim g) string length h) Check weather two strings are equal or not.

**Code**

import java.util.Scanner;

public class Q13 {

    public static void main(String[] args) {

        System.out.print("Enter the String1: ");

        Scanner sc = new Scanner(System.in);

        String string1 = sc.nextLine();

        System.out.print("Enter the String2: ");

        String string2 = sc.nextLine();

        System.out.println("The string1 is: " + string1);

        System.out.println("The string1 in uppercase is: " + string1.toUpperCase());

        System.out.println("The string1 in lowercase is: " + string1.toLowerCase());

        System.out.println("The total number of words in the string1 is: " + string1.split(" ").length);

        System.out.println("The substring of the string1 is: " + string1.substring(0, 5));

        if (string1.endsWith("Java")) {

            System.out.println("The string1 after concatenation is: " + string1.concat("StandardEdition"));

        }

        System.out.println("The string1 after trimming is: " + string1.trim());

        System.out.println("The length of the string1 is: " + string1.length());

        if (string1.equals(string2)) {

            System.out.println("The two strings are equal");

        } else {

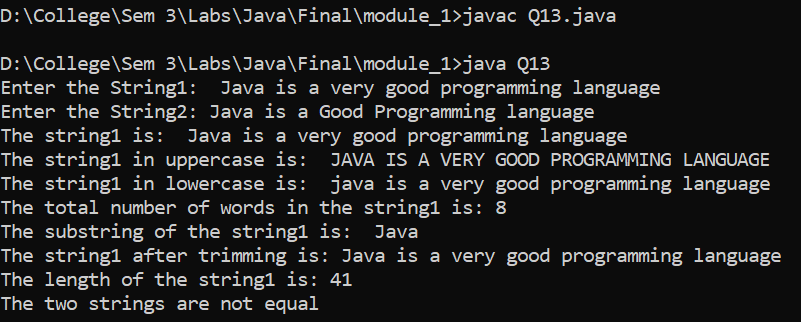
            System.out.println("The two strings are not equal");

        }

    }

}

**Input / Output**



Question 14:

Perform addition and multiplication of two matrix.

**Code**

public class Q14 {

    public static void main(String[] args) {

        int[][] a = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };

        int[][] b = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };

        int[][] c = new int[3][3];

        int[][] d = new int[3][3];

        System.out.println("Matrix A:");

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

            for (int j = 0; j < a[i].length; j++) {

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

            }

            System.out.println();

        }

        System.out.println("Matrix B:");

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

            for (int j = 0; j < b[i].length; j++) {

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

            }

            System.out.println();

        }

        for (int i = 0; i < 3; i++) {

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

                c[i][j] = a[i][j] + b[i][j];

                d[i][j] = a[i][j] \* b[i][j];

            }

        }

        System.out.println("Addition of two matrix is: ");

        for (int i = 0; i < 3; i++) {

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

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

            }

            System.out.println();

        }

        System.out.println("Multiplication of two matrix is: ");

        for (int i = 0; i < 3; i++) {

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

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

            }

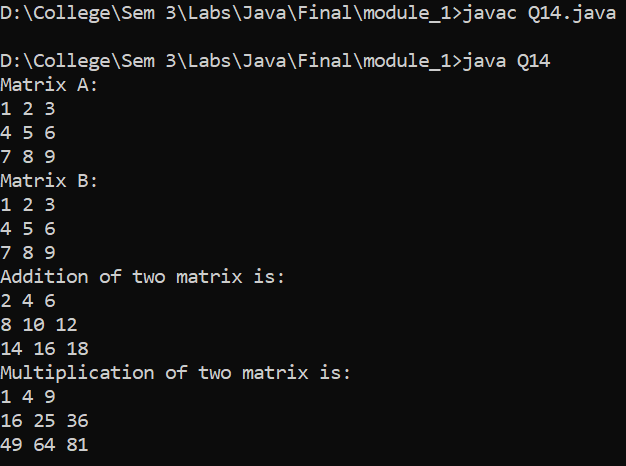
            System.out.println();

        }

    }

}

**Input / Output**



Question 15:

Write a program which will overload method calculateArea which calculates area of different shapes like circle, rectangle and square.

**Code**

public class Q15 {

    static double calculateArea (int r){

        return 3.14\*r\*r;

    }

    static int calculateArea (int l, int b){

        return l\*b;

    }

    public static void main(String[] args) {

        System.out.println("Area of Circle: " + calculateArea(5));

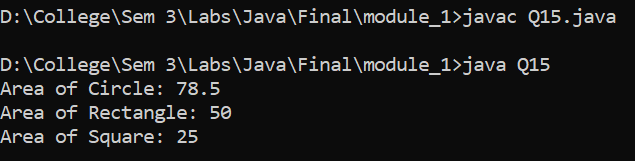
        System.out.println("Area of Rectangle: " + calculateArea(5, 10));

        System.out.println("Area of Square: " + calculateArea(5, 5));

    }

}

**Input / Output**

****

Question 16:

Write a program which will demonstrate the java.util.Arrays class methods like sort, copyOf, copyOfRange, fill, binarySearch, equals, toString etc.

**Code**

    import java.util.\*;

public class Q16 {

    public static void main(String[] args) {

        int[] a = { 1, 35, 22, 78, 23, 74, 65, 23, 54, 90 };

        int[] b = { 1, 35, 21, 73, 26, 74, 65, 23, 53, 97 };

        System.out.print("The Given Array is: ");

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

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

        }

        System.out.println();

        System.out.print("The Shorted Array is: ");

        Arrays.sort(a);

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

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

        }

        System.out.println();

        System.out.println("The Copied Array is: " + Arrays.toString(Arrays.copyOf(a, 5)));

        System.out.println("The Copied Range Array is: " + Arrays.toString(Arrays.copyOfRange(a, 2, 5)));

        Arrays.fill(a, 5);

        System.out.println("The Filled Array is: " + Arrays.toString(a));

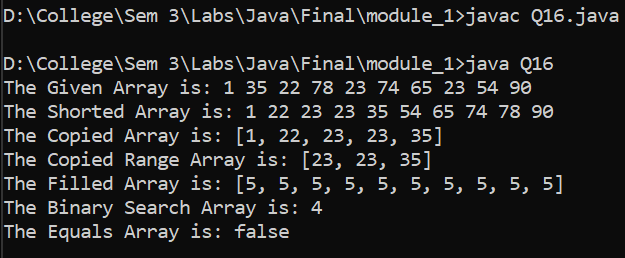
        System.out.println("The Binary Search Array is: " + Arrays.binarySearch(a, 5));

        System.out.println("The Equals Array is: " + Arrays.equals(a, b));

    }

}

**Input/ Output**

****

Question 17

Write a program which will make calculator application. Use Math class methods like pow, abs, max, min, exp, log, sqrt, cbrt, sin etc...

**Code**

import java.util.Scanner;

public class Q17 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int a = sc.nextInt();

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

        int b = sc.nextInt();

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

        String op = sc.next();

        switch (op) {

            case "+":

                System.out.println("The sum is: " + (a + b));

                break;

            case "-":

                System.out.println("The difference is: " + (a - b));

                break;

            case "\*":

                System.out.println("The product is: " + (a \* b));

                break;

            case "/":

                System.out.println("The quotient is: " + (a / b));

                break;

            case "%":

                System.out.println("The remainder is: " + (a % b));

                break;

            case "^":

                System.out.println("The power is: " + Math.pow(a, b));

                break;

            case "abs":

                System.out.println("The absolute value of a is: " + Math.abs(a));

                System.out.println("The absolute value of b is: " + Math.abs(b));

                break;

            case "max":

                System.out.println("The maximum value is: " + Math.max(a, b));

                break;

            case "min":

                System.out.println("The minimum value is: " + Math.min(a, b));

                break;

            case "exp":

                System.out.println("The exponential value of a is: " + Math.exp(a));

                System.out.println("The exponential value of b is: " + Math.exp(b));

                break;

            case "log":

                System.out.println("The logarithmic value of a is: " + Math.log(a));

                System.out.println("The logarithmic value of b is: " + Math.log(b));

                break;

            case "sqrt":

                System.out.println("The square root of a is: " + Math.sqrt(a));

                System.out.println("The square root of b is: " + Math.sqrt(b));

                break;

            case "cbrt":

                System.out.println("The cube root of a is: " + Math.cbrt(a));

                System.out.println("The cube root of b is: " + Math.cbrt(b));

                break;

            case "sin":

                System.out.println("The sine value of a is: " + Math.sin(a));

                System.out.println("The sine value of b is: " + Math.sin(b));

                break;

            case "cos":

                System.out.println("The cosine value of a is: " + Math.cos(a));

                System.out.println("The cosine value of b is: " + Math.cos(b));

                break;

            case "tan":

                System.out.println("The tangent value of a is: " + Math.tan(a));

                System.out.println("The tangent value of b is: " + Math.tan(b));

                break;

            default:

                System.out.println("Invalid operation");

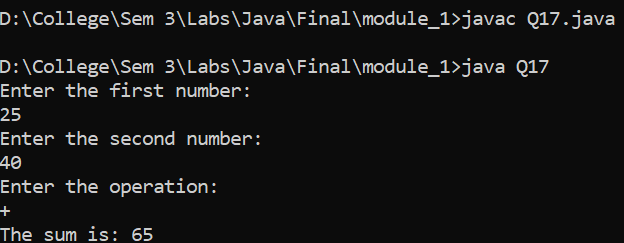
                break;

        }

    }

}

**Input / Output**

****

**Module 2**

**Topic**

**Public and Private access specifiers**

Question

An EmployeeDetail Class Which Take the Employ Detail

**Code**

class EmployeeDetail{

    private String eName;

    private int eId;

    private int Hour;

    private int eRate;

    static int count;

    public EmployeeDetail(String name, int id, int hour, int rate){

        set(name, id, hour, rate);

    }

    public void set(String name, int id, int hour, int rate){

        eName = name;

        eId = id + count++;

        Hour = hour;

        eRate = rate;

    }

    public void get(){

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

        System.out.println("Id: " + eId);

        System.out.println("Hour: " + Hour);

        System.out.println("Rate: " + eRate);

    }

}

public class Employee {

    public static void main(String[] args) {

        EmployeeDetail e1 = new EmployeeDetail("Dev", 1, 6, 20);

        EmployeeDetail e2 = new EmployeeDetail("Het", 2, 5, 10);

        EmployeeDetail e3 = new EmployeeDetail("Heet", 3, 6, 20);

        EmployeeDetail e4 = new EmployeeDetail("Mit", 4, 4, 15);

        e1.get();

        e2.get();

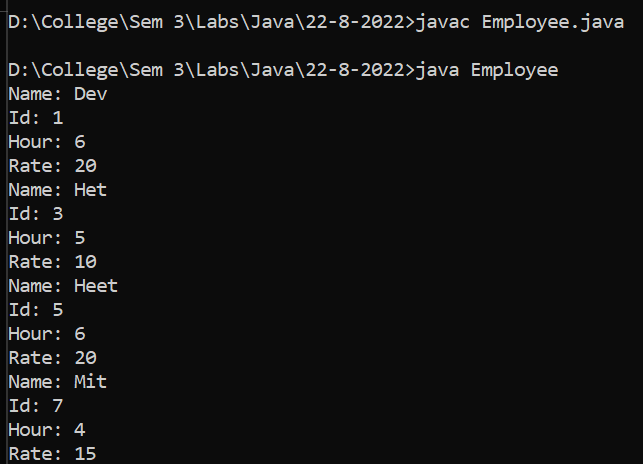
        e3.get();

        e4.get();

    }

}

**Input / Output**

****

**Topic**

**Constructer Overloading**

Question

A Product Class Which overload the constructor

**Code**

class ProductTest {

    int pID;

    String pName;

    int pRate;

    static int count;

    public ProductTest(String pName, int rate) {

        pID = 101 + count++;

        this.pName = pName;

        pRate = rate;

        System.out.println("Constructor called with " + pName + " and " + rate);

        System.out.println("Name: " + pName + " Rate: " + rate);

    }

    public ProductTest(String name) {

        pID = 101 + count++;

        pName = name;

        System.out.println("Constructor called with " + name);

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

    }

    public ProductTest() {

        this("NoName", 0);

        System.out.println("Default constructor called");

        System.out.println("NoName");

    }

}

public class Product {

    public static void main(String[] args) {

        ProductTest p1 = new ProductTest("Product1", 2);

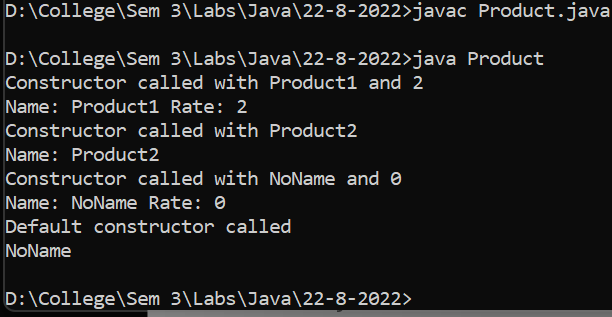
        ProductTest p2 = new ProductTest("Product2");

        ProductTest p3 = new ProductTest();

    }

}

**Input / Output**

****

**Topic**

**Access specifiers and Constructor overloading**

Question

Write program to create a class naming “Employee”.Having instance fields empId, hoursOfWork and rates. Provide default constructor, parameterized constructor, copy constructor, getter/ setter methods.Instance method naming getNetIncome() calculates salary of an Employee.Extends class SalesPerson from Employee. Add one more instance field to SalesPerson as salesAmount.Override getNetIncome. Provide necessary constructors and getter/setter methods.Call base class constructor from SalesPerson constructor.Write test class, which will create objects of above mentioned classes, copy the objects with copy constructor and call methods.Use appropriate access specifiers.(private, protected and public).

**Code**

class EmployeeDetails {

    String name;

    int eID;

    private int hours;

    private int rate;

    static int count;

    {

        eID = 101 + count++;

    }

    public EmployeeDetails(String name, int hours, int rate) {

        this.name = name;

        this.hours = hours;

        this.rate = rate;

    }

    public String getName() {

        return name;

    }

    public int getHours() {

        return hours;

    }

    public int getRate() {

        return rate;

    }

    public int findSalary() {

        int salary;

        if (getHours() > 180) {

            int extra = (getHours() - 180) \* getRate() / 2;

            salary = getHours() \* getRate() + extra;

        } else {

            salary = getHours() \* getRate();

        }

        return salary;

    }

    public void printInfo() {

        System.out.printf("EMP ID: %d\n", this.eID);

        System.out.printf("The name is : %s\n", getName());

        System.out.printf("The hours are : %d\n", getHours());

        System.out.printf("rate per hour is : %d\n", getRate());

        System.out.printf("The salary of the employee is :RS.%d\n", findSalary());

    }

}

public class Employee {

    public static void main(String[] args) {

        EmployeeDetails emp1 = new EmployeeDetails("Dev", 240, 50);

        emp1.printInfo();

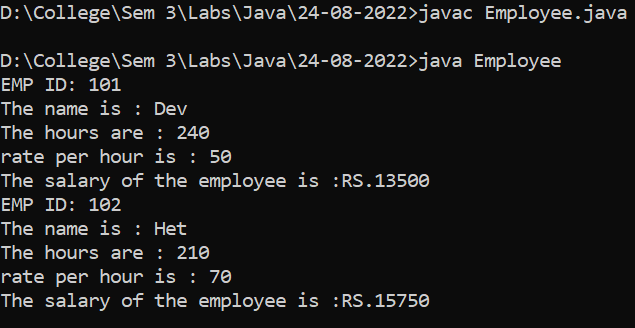
        EmployeeDetails emp2 = new EmployeeDetails("Het", 210, 70);

        emp2.printInfo();

    }

}

**Input / Output**

****

Question

A class of a Bank Account

**Code**

class AccountDetail {

    int AccNo;

    static int count = 1;

    {

        AccNo = 10000 + count++;

    }

    private int Balance;

    private String Name;

    static String BankName = "Sate Bank Of India";

    private long MoblieNo;

    public int getAccNo() {

        return AccNo;

    }

    public long getMoblieNo() {

        return MoblieNo;

    }

    public String getName() {

        return Name;

    }

    public int getBalance() {

        return Balance;

    }

    AccountDetail(String name, long mobile) {

        Name = name;

        MoblieNo = mobile;

    }

    AccountDetail(String name) {

        Name = name;

    }

    public String Deposite(int amount) {

        Balance = Balance + amount;

        return ("You Deposite: " + amount + " rs\n Current Bank Balance for Account Number '" + getAccNo() + "' is "

                + getBalance() + " rs");

    }

    public String Withdraw(int amount) {

        Balance = Balance - amount;

        return ("You Withdraw: " + amount + " rs\n Current Bank Balance for Account Number '" + getAccNo() + "' is "

                + getBalance() + " rs");

    }

}

public class Account {

    public static int Total(AccountDetail arr[]){

        int sum=0;

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

            sum+=arr[i].getBalance();

        }

        return sum;

    }

    public static void main(String[] args) {

        AccountDetail AccHolder1 = new AccountDetail("Dev", 9999999999l);

        AccountDetail AccHolder2 = new AccountDetail("Mit");

        AccountDetail AccHolder3 = new AccountDetail("Neel", 9898989898l);

        AccountDetail AccHolder4 = new AccountDetail("Ved", 98942544334l);

        AccountDetail AccHolder5 = new AccountDetail("Roy", 9845436654l);

        System.out.println(AccHolder1.Deposite(110));

        System.out.println(AccHolder2.Deposite(800));

        System.out.println(AccHolder3.Deposite(300));

        System.out.println(AccHolder4.Deposite(500));

        System.out.println(AccHolder5.Deposite(100));

        System.out.println(AccHolder1.Withdraw(10));

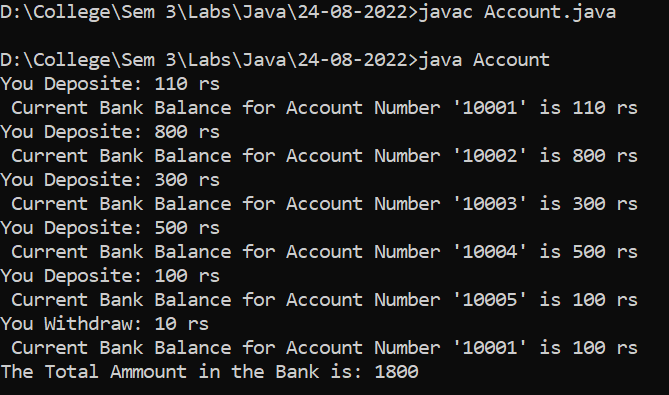
        AccountDetail[] dtl = {AccHolder1,AccHolder2,AccHolder3,AccHolder4,AccHolder5 };

        System.out.println("The Total Ammount in the Bank is: "+Account.Total(dtl));

    }

}

**Input / Output**

****

**Topic**

**Abstract Class, Inheritance, Runtime Polymorphism**

Question

Write a class naming BankAccount.Provide three instance method withdraw, deposit and transfer. Instance fields accid, balance.Extends SavingsAccount and CurrentAccount class from BankAccount.Override withdraw method in both these classes. Provide necessary default constructor, parameterized constructor, copy constructor, getter/ setter methods in all three classes. Write one test class to test above mentioned classes.

**Code**

class Account {

    int accNo;

    static int count;

    {

        accNo=10001+count++;

    }

    private int  balance;

    private long mobileNo;

    private String Name;

    static String bankname="HDFC";

    public Account(){

        System.out.println("Account created without any information");

    }

    public Account(String name){

        System.out.println("Account created but you have not given the mobile number So account can not be linked " +

                "mobile no.");

    }

    public Account(String name,long mobileNo) {

        this.Name = name;

        this.mobileNo = mobileNo;

    }

    public int getAccNo() {

        return accNo;

    }

    public String getName() {

        return Name;

    }

    public void setName(String name){

        this.Name=name;

    }

    public long getMobileNo(){

        return mobileNo;

    }

    public void setMobileNo(long mobileNo){

        this.mobileNo=mobileNo;

    }

    public int getBalance() {

        return balance;

    }

    public String Deposit(int dep\_money){

        this.balance+=dep\_money;

        return (dep\_money+"RS"+" debited to your account\n");

    }

    public String checkBalance(){

        return ("Your balance is : "+getBalance()+"\n");

    }

    public String withdraw(int money){

        this.balance-=money;

        return ("You have withdraw "+money+"RS from your account\n");

    }

    public void getAccinfo(){

        System.out.println("Bank name: "+bankname);

        System.out.println("Account holder name: "+getName());

        System.out.println("Mobile No."+getMobileNo());

        System.out.println("Account number :"+getAccNo());

        System.out.println("Balance : "+getBalance());

    }

}

class SavingAccount extends Account{

    int interestRate;

    public SavingAccount(){

        System.out.println("Savings Account created");

    }

    public SavingAccount(String name){

        System.out.println("Savings Account created ");

        name = this.getName();

        System.out.printf("Account holder name:%s", name);

    }

    public SavingAccount(String name,long number,int interestRate){

        super.setName(name);

        super.setMobileNo(number);

        this.interestRate=interestRate;

    }

    public String withdraw(int money){

        super.withdraw(money);

        if(super.getBalance()-money>1000){

            return ( money+"Rs has debited successfully from your account");

        }

        else{

            return ("Insufficient balance you can not withdraw money");

        }

    }

}

class Acc{

    public static void main(String[] args) {

        Account a1=new Account("Dev Sapariya", 999999999L);

        System.out.println(a1.checkBalance());

        System.out.println(a1.Deposit(10000));

        System.out.println(a1.checkBalance());

        System.out.println(a1.withdraw(500));

        a1.getAccinfo();

        SavingAccount s1=new SavingAccount("Mit",8888888888L,7);

        System.out.println(s1.checkBalance());

        System.out.println(s1.Deposit(10000));

        System.out.println(s1.checkBalance());

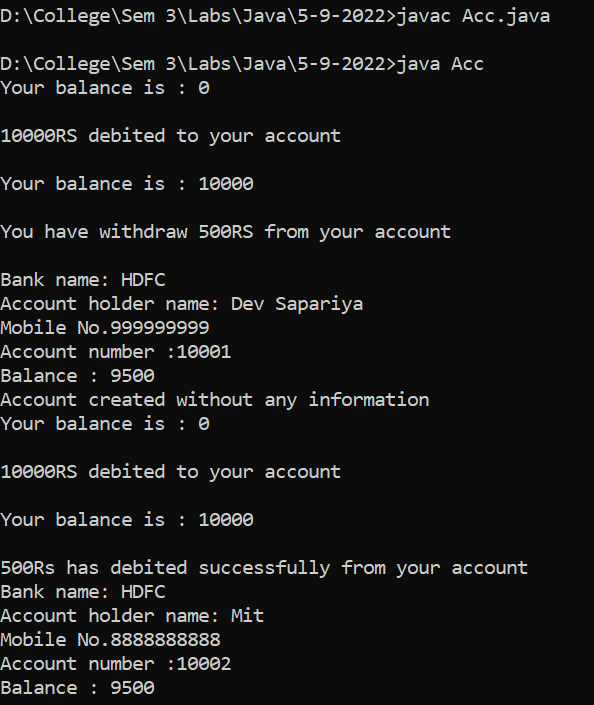
        System.out.println(s1.withdraw(500));

        s1.getAccinfo();

    }

}

**Input / Output**

****

Question

Describe abstract class called TwoDShape having two instance fields length and width.

Create default constructor, parameterized constructor, copy constructor. Getter and setter method. abstract  int getArea();

It has three subclasses say Triangle, Rectangle, and Square.

Override getArea() method in these three subclasses to calculate area for specific object i.e. getArea() of Triangle subclass should calculate area of triangle etc. Same for Rectangle and Square.

**Code**

abstract class TwoDShape {

    int length;

    int width;

    TwoDShape() {

        length = 0;

        width = 0;

    }

    TwoDShape(int length, int width) {

        this.length = length;

        this.width = width;

    }

    TwoDShape(TwoDShape t) {

        length = t.length;

        width = t.width;

    }

    public void setLength(int length) {

        this.length = length;

    }

    public void setWidth(int width) {

        this.width = width;

    }

    abstract float getArea();

}

class Rectangle extends TwoDShape {

    public Rectangle() {

        length = 10;

        width = 10;

    }

    public Rectangle(int l, int w) {

        super(l, w);

    }

    public Rectangle(TwoDShape r) {

        super(r);

    }

    public float getArea() {

        return length \* width;

    }

}

class Triangle extends TwoDShape {

    public Triangle(int l, int w) {

        super(l, w);

    }

    public float getArea() {

        return (float) (0.5 \* length \* width);

    }

}

public class Question\_area {

    public static void main(String[] args) {

        TwoDShape t = new Rectangle(5, 6);

        TwoDShape t1 = new Rectangle(t);

        Rectangle r = new Rectangle(4, 6);

        TwoDShape tri = new Triangle(3, 6);

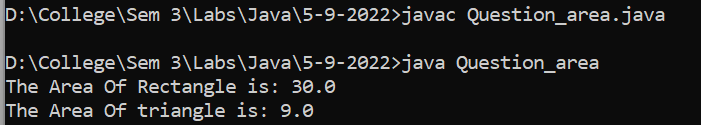
        System.out.println("The Area Of Rectangle is: " + t.getArea());

        System.out.println("The Area Of triangle is: " + tri.getArea());

    }

}

**Input/Output**

****

**Topic**

**Copy Constructor, Return the Object**

Question

A Measurement class which give the Difference of the measurement

**Code**

abstract class Measurement {

    protected int unit1;

    protected int unit2;

    protected int foot1;

    protected int foot2;

    protected int inch1;

    protected int inch2;

    Measurement() {

        unit1 = 10;

        unit2 = 20;

    }

    Measurement(int foot1, int inch1, int foot2, int inch2) {

        this.foot1 = foot1;

        this.foot2 = foot2;

        this.inch1 = inch1;

        this.inch2 = inch2;

        unit1 = (foot1 \* 12) + inch1;

        unit2 = (foot2 \* 12) + inch2;

    }

    Measurement(Measurement m) {

        foot1 = m.foot1;

        foot2 = m.foot2;

        inch1 = m.inch1;

        inch2 = m.inch2;

        unit1 = (foot1 \* 12) + inch1;

        unit2 = (foot2 \* 12) + inch2;

    }

    public int getUnit1() {

        return unit1;

    }

    public int getUnit2() {

        return unit2;

    }

    public void setUnit1(int unit1) {

        this.unit1 = unit1;

    }

    public void setUnit2(int unit2) {

        this.unit2 = unit2;

    }

    abstract Measurement getDifference(Measurement m);

}

class MeasurementPrint extends Measurement {

    int foot;

    int inch;

    MeasurementPrint(int foot, int inch) {

        this.foot = foot;

        this.inch = inch;

    }

    void printfootinch() {

        System.out.println("foot :" + foot + " inch: " + inch);

    }

    @Override

    Measurement getDifference(Measurement m) {

        return null;

    }

}

class HeightMeasurement extends Measurement {

    HeightMeasurement(int foot1, int inch1, int foot2, int inch2) {

        super(foot1, inch1, foot2, inch2);

    }

    HeightMeasurement(Measurement m) {

        super(m);

    }

    @Override

    Measurement getDifference(Measurement m) {

        int diff = m.unit2 - m.unit1;

        int foot = (diff) / 12;

        int inch = diff % 12;

        Measurement m6 = new MeasurementPrint(foot, inch);

        return m6;

    }

}

public class Question\_measurement {

    public static void main(String[] args) {

        Measurement m = new HeightMeasurement(6, 2, 9, 3);

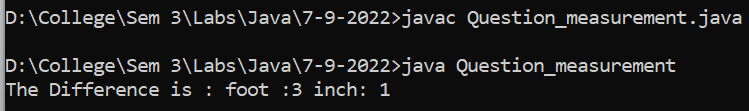
        System.out.print("The Difference is : ");

        ((MeasurementPrint) m.getDifference(m)).printfootinch();

    }

}

**Input / Output**

****

**Topic**

**Interface**

Question

Interface Example

**Code**

interface Interface1{

    public abstract  void printMethod1();

}

class Testclass1 implements Interface1{

    private  int number;

    public  Testclass1(){

        number=20;

    }

    public  Testclass1(int n){

        number=n;

    }

    public  int getNumber(){

        return  number;

    }

    public  void  setNumber(int n){

        number=n;

    }

    public void printMethod1(){

        System.out.println("TestCalss1 : "+number);

    }

}

public class InterfaceTesting {

    public static void main(String[] args) {

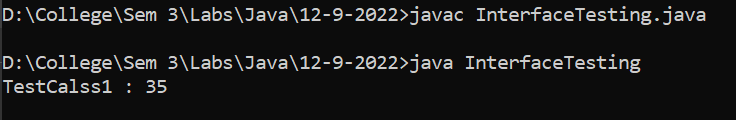
            Interface1 i1= new Testclass1(35);

            i1.printMethod1();

    }

}

**Input / Output**

****

Question

A Class Describe As Below

abstact class Residency

    instance fields :name, rNumber, area, unit\_rate

    provide all overloaded constructor including copy constructor

    provide necessary getter/setter method.

    abstract long getPriceOfResidency();

   override toString() and equals() method from Object class.

create abstract class naming Semi\_FurnishedResidency extends Residency.

   instance field - furnitureCharge, parkingCharge

   create class LuxuriousResidency extends Residency.

   instance field - amenityCharge

   override long getPriceOfResidency().

   create interface naming Rentable -

   abstract public int getRent();

   create TwoBHKResidency extends Semi\_FurnishedResidency implements Rentable.

   override long getPriceOfResidency().

   override int getRent().

**Code**

abstract class Residency {

    public String name;

    private int rNumber;

    private int area;

    private int unit\_rate;

    public Residency() {

        name = null;

        rNumber = 0;

        area = 0;

        unit\_rate = 0;

    }

    public Residency(String name, int rNumber, int area, int unit\_rate) {

        this.name = name;

        this.rNumber = rNumber;

        this.area = area;

        this.unit\_rate = unit\_rate;

    }

    Residency(Residency r) {

        this(r.name, r.rNumber, r.area, r.unit\_rate);

    }

    public int getArea() {

        return area;

    }

    public int getrNumber() {

        return rNumber;

    }

    public int getUnit\_rate() {

        return unit\_rate;

    }

    public String getName() {

        return name;

    }

    public void setArea(int area) {

        this.area = area;

    }

    public void setName(String name) {

        this.name = name;

    }

    public void setUnit\_rate(int unit\_rate) {

        this.unit\_rate = unit\_rate;

    }

    public void setrNumber(int rNumber) {

        this.rNumber = rNumber;

    }

    abstract long getPriceOfResidency();

}

abstract class Semi\_FurnishedResidency extends Residency {

    int furnitureCharge = 1000;

    int parkingCharge = 99;

    int price;

    Semi\_FurnishedResidency() {

        super();

    }

    Semi\_FurnishedResidency(String name, int rNumber, int area, int unit\_rate) {

        super.setName(name);

        super.setArea(area);

        super.setrNumber(rNumber);

        super.setUnit\_rate(unit\_rate);

    }

}

class LuxuriousResidency extends Residency {

    int amenityCharge = 66600;

    int price;

    @Override

    long getPriceOfResidency() {

        price = (getUnit\_rate() \* getArea()) + amenityCharge;

        return price;

    }

    LuxuriousResidency() {

        super();

    }

    LuxuriousResidency(String name, int rNumber, int area, int unit\_rate) {

        super.setName(name);

        super.setArea(area);

        super.setrNumber(rNumber);

        super.setUnit\_rate(unit\_rate);

    }

}

interface Rentable {

    abstract public long getRent();

}

class TwoBHKResidency extends Semi\_FurnishedResidency implements Rentable {

    long rent;

    public TwoBHKResidency(String name, int rNumber, int area, int unit\_rate) {

        super(name, rNumber, area, unit\_rate);

    }

    @Override

    public long getRent() {

        rent = (getPriceOfResidency() \* 5) / 100;

        return rent;

    }

    @Override

    long getPriceOfResidency() {

        price = (getUnit\_rate() \* getArea()) + furnitureCharge + parkingCharge;

        return price;

    }

}

public class Residency\_main {

    public static void main(String[] args) {

        LuxuriousResidency r1 = new LuxuriousResidency("PYTHON", 1,2000,63);

        System.out.println("The price of Luxurious Residency is: "+r1.getPriceOfResidency());

        Semi\_FurnishedResidency r2 = new TwoBHKResidency("JAVA", 2, 1000, 50);

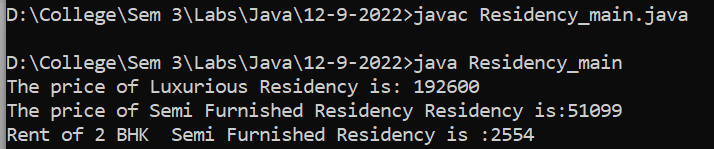
        System.out.println("The price of Semi Furnished Residency Residency is:"+r2.getPriceOfResidency());

        System.out.println("Rent of 2 BHK  Semi Furnished Residency is :"+((Rentable) r2).getRent());

    }

}

**Input / Output**

****

**Topic**

**Pass by Value and Variable Argument**

Pass By Value Example

class Account {

    private int accNo;

    private double balance;

    public Account(int an, double am) {

        accNo = an;

        balance = am;

    }

    public void deposit(double amount) {

        balance = balance + amount;

    }

    public void transfer(Account ac1, Account ac2) {

    }

    public double getBalance() {

        return balance;

    }

}

public class JavaPassByValue {

    public static void testPassByValue(Account ac, int amount) {

        amount = amount + 1000;

        ac.deposit(7000);

        System.out.println("Balance in method :" + ac.getBalance());

        ac = new Account(102, 5000);

        ac.deposit(2000);

        System.out.println("New Acc Balance in method :" + ac.getBalance());

        System.out.println("Amount Value in function : " + amount);

    }

    public static void main(String[] args) {

        int amount = 2000;

        Account ac = new Account(101, amount);

        ac.deposit(4000);

        testPassByValue(ac, amount);

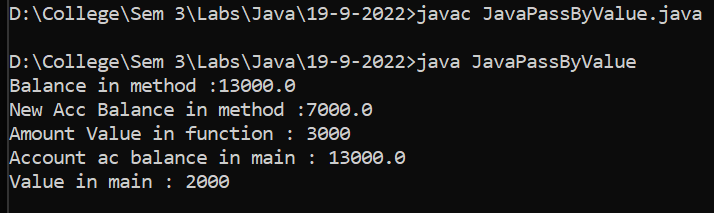
        System.out.println("Account ac balance in main : " + ac.getBalance());

        System.out.println("Value in main : " + amount);

    }

}

**Input / Output**

****

Variable Argument Example

**Code**

class VarArgTest {

    public static double average(double first, double second, double... remaining) {

        double total = first + second;

        for (double value : remaining)

            total += value;

        return total / (remaining.length + 2);

    }

    public static void main(String[] args) {

        System.out.printf("Average = %.3f%n", average(12.3, 13.7, 11.9, 19.8, 14.1, 17.7));

    }

    public static int maximum(String SubName, int DivisionNo, int... marks) {

        int max = marks[0];

        for (int num : marks) {

            if (num > max) {

                max = num;

            }

        }

        return max;

    }

}

public class VarArgs {

    public static void main(String[] args) {

        double avg = VarArgTest.average(10, 20, 30, 40, 50, 60, 70);

        System.out.println(avg);

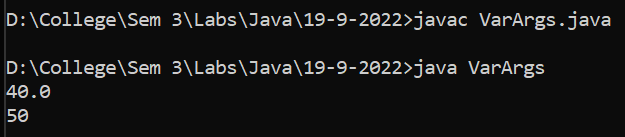
        int max = VarArgTest.maximum("Maths", 5, 20, 30, 40, 50);

        System.out.println(max);

    }

}

**Input / Output**

****

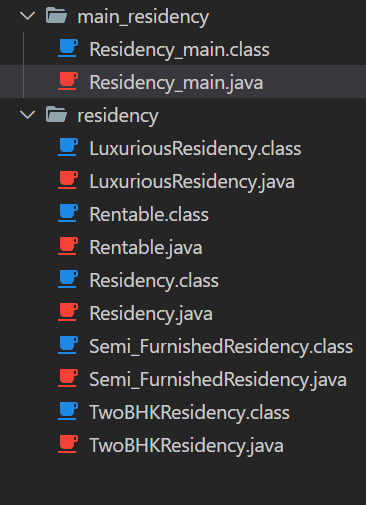
**Topic**

**Package**

Question

Example of using package in Residency Class

**File Structure**

****

**Code**

**Residency\_main.java**

package main\_residency;

import residency.\*;

public class Residency\_main {

    public static void main(String[] args) {

        LuxuriousResidency r1 = new LuxuriousResidency("PYTHON", 1,2000,63);

        System.out.println("The price of Luxurious Residency is: "+r1.getPriceOfResidency());

        Semi\_FurnishedResidency r2 = new TwoBHKResidency("JAVA", 2, 1000, 50);

        System.out.println("The price of Semi Furnished Residency Residency is:"+r2.getPriceOfResidency());

        System.out.println("Rent of 2 BHK  Semi Furnished Residency is :"+((Rentable) r2).getRent());

    }

}

**LuxuriousResidency.java**

package residency;

public class LuxuriousResidency extends Residency {

    int amenityCharge = 66600;

    int price;

    @Override

    public

    long getPriceOfResidency() {

        price = (getUnit\_rate() \* getArea()) + amenityCharge;

        return price;

    }

    LuxuriousResidency() {

        super();

    }

    public LuxuriousResidency(String name, int rNumber, int area, int unit\_rate) {

        super.setName(name);

        super.setArea(area);

        super.setrNumber(rNumber);

        super.setUnit\_rate(unit\_rate);

    }

}

**Rentable.java**

package residency;

public interface Rentable {

    abstract public long getRent();

}

**Residency.java**

package residency;

abstract class Residency {

    public String name;

    private int rNumber;

    private int area;

    private int unit\_rate;

    public Residency() {

        name = null;

        rNumber = 0;

        area = 0;

        unit\_rate = 0;

    }

    public Residency(String name, int rNumber, int area, int unit\_rate) {

        this.name = name;

        this.rNumber = rNumber;

        this.area = area;

        this.unit\_rate = unit\_rate;

    }

    Residency(Residency r) {

        this(r.name, r.rNumber, r.area, r.unit\_rate);

    }

    public int getArea() {

        return area;

    }

    public int getrNumber() {

        return rNumber;

    }

    public int getUnit\_rate() {

        return unit\_rate;

    }

    public String getName() {

        return name;

    }

    public void setArea(int area) {

        this.area = area;

    }

    public void setName(String name) {

        this.name = name;

    }

    public void setUnit\_rate(int unit\_rate) {

        this.unit\_rate = unit\_rate;

    }

    public void setrNumber(int rNumber) {

        this.rNumber = rNumber;

    }

    abstract long getPriceOfResidency();

}

**Semi\_FurnishedResidency.java**

package residency;

public class Semi\_FurnishedResidency extends Residency{

    int furnitureCharge = 1000;

    int parkingCharge = 99;

    int price;

    @Override

    public

    long getPriceOfResidency() {

        price = (getUnit\_rate() \* getArea()) + furnitureCharge + parkingCharge;

        return price;

    }

    Semi\_FurnishedResidency() {

        super();

    }

    public Semi\_FurnishedResidency(String name, int rNumber, int area, int unit\_rate) {

        super.setName(name);

        super.setArea(area);

        super.setrNumber(rNumber);

        super.setUnit\_rate(unit\_rate);

    }

}

**TwoBHKResidency.java**

package residency;

public class TwoBHKResidency extends Semi\_FurnishedResidency implements Rentable {

    long rent;

    public TwoBHKResidency(String name, int rNumber, int area, int unit\_rate) {

        super(name, rNumber, area, unit\_rate);

    }

    @Override

    public long getRent() {

        rent = (getPriceOfResidency() \* 5) / 100;

        return rent;

    }

    @Override

    public long getPriceOfResidency() {

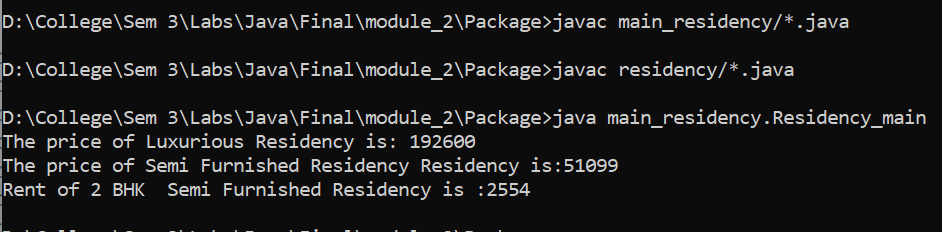
        price = (getUnit\_rate() \* getArea()) + furnitureCharge + parkingCharge;

        return price;

    }

}

**Input / Output**

****

**Module 3**

**Question 1:**

Write a program to read file, encrypt it (encryption key is +3), and store encrypted data to another file.(Use FileInputStream/OutputStream).

**Code**

import java.io.\*;

class Qu1 {

    public static void main(String args[]) throws IOException {

        FileInputStream fin = new FileInputStream("Qu1.java");

        FileOutputStream fout = new FileOutputStream("Qu1.txt");

        int i;

        while ((i = fin.read()) != -1) {

            fout.write(i + 3);

        }

        fin.close();

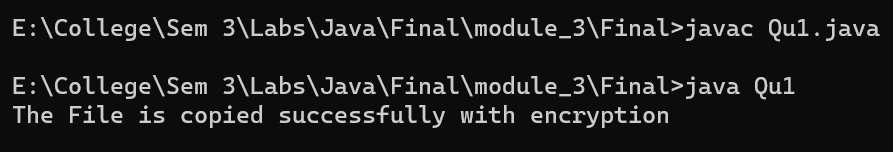
        fout.close();

        System.out.println("The File is copied successfully with encryption");

    }

}

**Output**

****

**Qu1.txt**

lpsruw#mdyd1lr1->

fodvv#Tx4#~

####sxeolf#vwdwlf#yrlg#pdlq+Vwulqj#dujv^`,#wkurzv#LRH{fhswlrq#~

########IlohLqsxwVwuhdp#ilq#@#qhz#IlohLqsxwVwuhdp+%Tx41mdyd%,>

########IlohRxwsxwVwuhdp#irxw#@#qhz#IlohRxwsxwVwuhdp+%Tx41w{w%,>

########lqw#l>

########zkloh#++l#@#ilq1uhdg+,,#$@#04,#~

############irxw1zulwh+l#.#6,>

########�

########ilq1forvh+,>

########irxw1forvh+,>

########V|vwhp1rxw1sulqwoq+%Wkh#Iloh#lv#frslhg#vxffhvvixoo|#zlwk#hqfu|swlrq%,>

####�

�

**Question 2:**

Write a program, which reads encrypted file generated in 1st program, decrypt it, print it to the console.(Use FileInputStream).

**Code**

import java.io.\*;

public class Qu2 {

    public static void main(String[] args) {

        try {

            FileInputStream fin = new FileInputStream("Qu1.txt");

            int i;

            while ((i = fin.read()) != -1) {

                System.out.print((char) (i - 3));

            }

            fin.close();

        } catch (Exception e) {

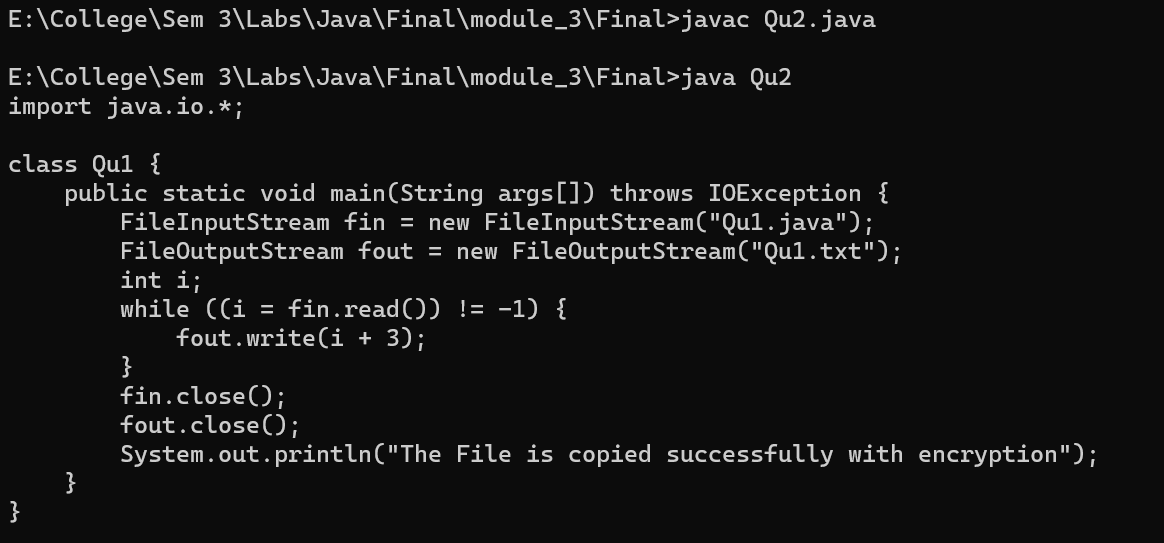
            System.out.println(e);

        }

    }

}

**Output**

****

**Question 3:**

Write a program to Copy the data of a file, file name provided from command line argument, to another file.(Use Buffered I/O Stream).

**Code**

import java.io.\*;

public class Qu3 {

    public static void main(String[] args) throws IOException {

        FileInputStream fin = new FileInputStream(args[0]);

        FileOutputStream fos = new FileOutputStream(args[1]);

        BufferedInputStream buffin = new BufferedInputStream(fin);

        BufferedOutputStream buffout = new BufferedOutputStream(fos);

        int i;

        while ((i = buffin.read()) != -1) {

            buffout.write(i);

        }

        buffin.close();

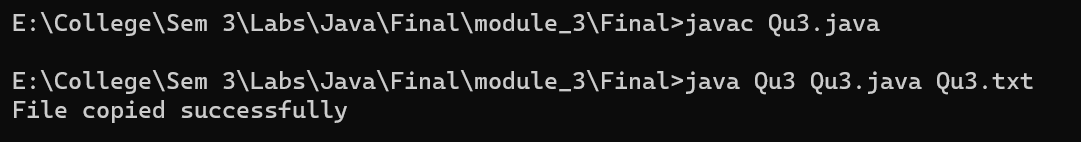
        buffout.close();

        System.out.println("File copied successfully");

    }

}

**Output**

****

**Qu3.txt**

import java.io.\*;

public class Qu3 {

public static void main(String[] args) throws IOException {

FileInputStream fin = new FileInputStream(args[0]);

FileOutputStream fos = new FileOutputStream(args[1]);

BufferedInputStream buffin = new BufferedInputStream(fin);

BufferedOutputStream buffout = new BufferedOutputStream(fos);

int i;

while ((i = buffin.read()) != -1) {

buffout.write(i);

}

buffin.close();

buffout.close();

System.out.println("File copied successfully");

}

}

**Question 4:**

Write a ProductManagement program, which will store 5 objects of Product class to a file name “ProductDetails.dat”.(Use ObjectOutputStream)

**Code**

import java.io.\*;

class Product implements Serializable

{

    int pid;

    String pname;

    double price;

    Product(int pid,String pname,double price)

    {

        this.pid=pid;

        this.pname=pname;

        this.price=price;

    }

    public String toString()

    {

        return pid+" "+pname+" "+price;

    }

}

public class Qu4

{

    public static void main(String args[]) throws IOException

    {

        Product p1=new Product(1,"Apple",100);

        Product p2=new Product(2,"Samsung",200);

        Product p3=new Product(3,"Vivo",3000);

        Product p4=new Product(4,"Oppo",400);

        Product p5=new Product(5,"Nothing",500);

        FileOutputStream fout=new FileOutputStream("ProductDetails.dat");

        ObjectOutputStream out=new ObjectOutputStream(fout);

        out.writeObject(p1);

        out.writeObject(p2);

        out.writeObject(p3);

        out.writeObject(p4);

        out.writeObject(p5);

        out.close();

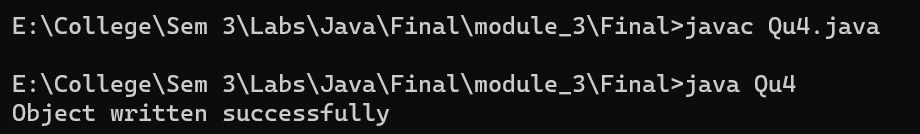
        fout.close();

        System.out.println("Object written successfully");

    }

}

**Output**

****

**Question 5:**

Write a program, Which will read file “ProductDetails.dat” (created by above program).Print the details of the object having highest price and lowest price. (Use ObjectInputStream).

**Code**

import java.io.\*;

public class Qu5 {

    public static void main(String[] args) {

        try {

            FileInputStream fin = new FileInputStream("ProductDetails.dat");

            ObjectInputStream in = new ObjectInputStream(fin);

            Product p1 = (Product) in.readObject();

            Product p2 = (Product) in.readObject();

            Product p3 = (Product) in.readObject();

            Product p4 = (Product) in.readObject();

            Product p5 = (Product) in.readObject();

            Product[] p = {p1, p2, p3, p4, p5};

            double max = p[0].price;

            double min = p[0].price;

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

                if (p[i].price > max) {

                    max = p[i].price;

                }

                if (p[i].price < min) {

                    min = p[i].price;

                }

            }

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

                if (p[i].price == max) {

                    System.out.println("Highest Price: " + p[i]);

                }

                if (p[i].price == min) {

                    System.out.println("Lowest Price: " + p[i]);

                }

            }

            in.close();

            fin.close();

        } catch (Exception e) {

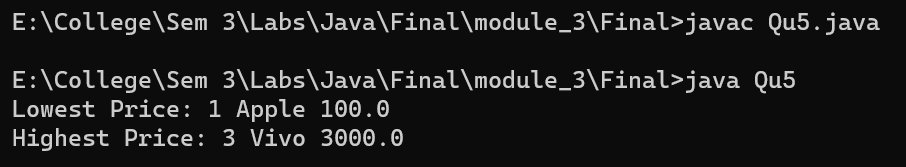
            System.out.println(e);

        }

    }

}

**Output**

****

**Question 6:**

Write a program which will read a text file and print the count of total number of Lines, Words and Characters in it. (Use BufferedReader).

**Code**

import java.io.\*;

public class Qu6 {

    public static void main(String[] args) {

        try {

            BufferedReader br = new BufferedReader(new FileReader("Qu6.java"));

            int lines = 0, words = 0, chars = 0;

            String s;

            while ((s = br.readLine()) != null) {

                lines++;

                String[] w = s.split(" ");

                words += w.length;

                chars += s.length();

            }

            System.out.println("Lines: " + lines);

            System.out.println("Words: " + words);

            System.out.println("Characters: " + chars);

            br.close();

        } catch (Exception e) {

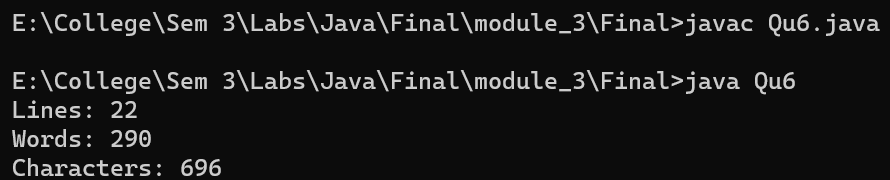
            System.out.println(e);

        }

    }

}

**Output**

****

**Question 7:**

Write a program to read a text file and copy it’s content in uppercase form to another file.(Use BufferedWriter\Reader).

**Code**

import java.io.\*;

public class Qu7 {

    public static void main(String[] args) {

        try {

            BufferedReader br = new BufferedReader(new FileReader("Qu7.java"));

            BufferedWriter bw = new BufferedWriter(new FileWriter("Qu7.txt"));

            String s;

            while ((s = br.readLine()) != null) {

                bw.write(s.toUpperCase());

                bw.newLine();

            }

            br.close();

            bw.close();

        } catch (Exception e) {

            System.out.println(e);

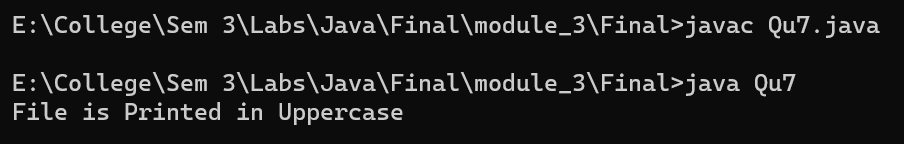
        }

        System.out.println("File is Printed in Uppercase");

    }

}

**Output**

****

**Qu7.txt**

IMPORT JAVA.IO.\*;

PUBLIC CLASS QU7 {

PUBLIC STATIC VOID MAIN(STRING[] ARGS) {

TRY {

BUFFEREDREADER BR = NEW BUFFEREDREADER(NEW FILEREADER("QU7.JAVA"));

BUFFEREDWRITER BW = NEW BUFFEREDWRITER(NEW FILEWRITER("QU7.TXT"));

STRING S;

WHILE ((S = BR.READLINE()) != NULL) {

BW.WRITE(S.TOUPPERCASE());

BW.NEWLINE();

}

BR.CLOSE();

BW.CLOSE();

} CATCH (EXCEPTION E) {

SYSTEM.OUT.PRINTLN(E);

}

SYSTEM.OUT.PRINTLN("FILE IS PRINTED IN UPPERCASE");

}

}

**Question 8:**

Write a program to copy content of a file while removing duplicates lines.

**Code**

import java.io.\*;

import java.util.HashSet;

public class Qu8 {

    public static void main(String[] args) throws IOException {

        // PrintWriter pw = new PrintWriter("Qu81.txt");

        FileOutputStream fos = new FileOutputStream("Qu8-1.txt");

        BufferedOutputStream buffout = new BufferedOutputStream(fos);

        BufferedReader br = new BufferedReader(new FileReader("Qu8.txt"));

        String line = br.readLine();

        HashSet<String> hs = new HashSet<String>();

        while (line != null) {

            hs.add(line);

            line = br.readLine();

        }

        for (String s : hs) {

            buffout.write(s.getBytes());

            buffout.write("\n".getBytes());

        }

        br.close();

        buffout.close();

        System.out.println("File operation performed successfully");

    }

}

**Qu8.txt**

this is line 1

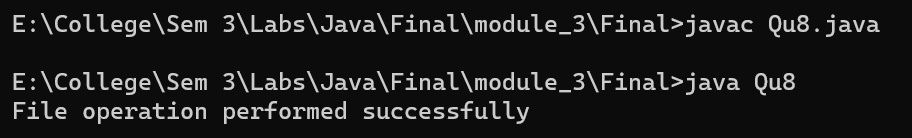
this is line 1

this is line 2

this is line 3

this is line 3

**Output**

****

**Qu8-1.txt**

this is line 3

this is line 2

this is line 1

**Question 9:**

Write an Exception handling program, which will handle RuntimeException, ArrayIndexOutOfBoundsException, NumberFormatException, ArithmeticException, NullPointerException.(Use Multiple catch with single try block).

**Code**

public class Qu9 {

    public static void main(String[] args) {

        try {

            int a = Integer.parseInt(args[0]);

            int b = Integer.parseInt(args[1]);

            int c = a / b;

            System.out.println(c);

        } catch (ArithmeticException e) {

            System.out.println("ArithmeticException: " + e);

        } catch (ArrayIndexOutOfBoundsException e) {

            System.out.println("ArrayIndexOutOfBoundsException: " + e);

        } catch (NumberFormatException e) {

            System.out.println("NumberFormatException: " + e);

        } catch (NullPointerException e) {

            System.out.println("NullPointerException: " + e);

        } catch (RuntimeException e) {

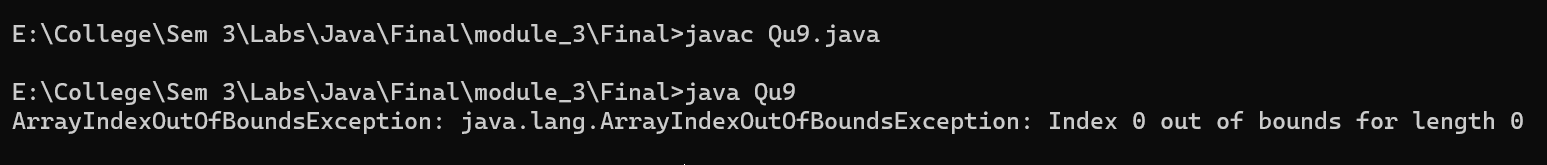
            System.out.println("RuntimeException: " + e);

        }

    }

}

**Output**

****

**Question 10:**

Write a program, to demonstrate nested try-catch-finally structure.

**Code**

//Write a program, to demonstrate nested try-catch-finally structure.

import java.io.\*;

public class Qu10 {

    public static void main(String[] args) {

        try {

            try {

                int a = 10 / 0;

            } catch (ArithmeticException e) {

                System.out.println("Arithmetic Exception");

            }

            try {

                int[] a = new int[5];

                a[10] = 10;

            } catch (ArrayIndexOutOfBoundsException e) {

                System.out.println("ArrayIndexOutOfBounds Exception");

            }

        } catch (Exception e) {

            System.out.println("Parent Exception");

        } finally {

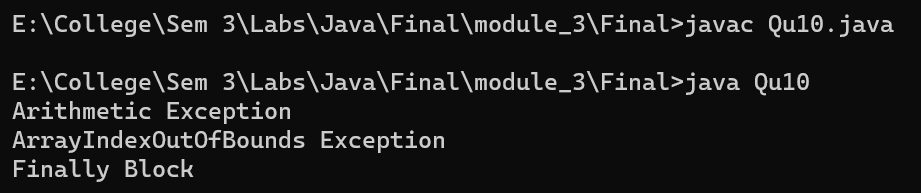
            System.out.println("Finally Block");

        }

    }

}

**Output**

****

**Question 11:**

Write a program, to create and handle user defined Unchecked Exception – InvalidBoxException which will be thrown from the constructor of the Box class, when either length or width or height of Box is less than zero.

**Code**

class InvalidBoxException extends Exception {

    InvalidBoxException(String s) {

        super(s);

    }

}

class Box {

    int length;

    int width;

    int height;

    Box(int l, int w, int h) throws InvalidBoxException {

        if (l < 0 || w < 0 || h < 0) {

            throw new InvalidBoxException("invaild input");

        } else {

            System.out.println("Done");

        }

    }

}

public class Qu11 {

    public static void main(String[] args) {

        try {

            Box b = new Box(1, -22, 4);

        } catch (InvalidBoxException e) {

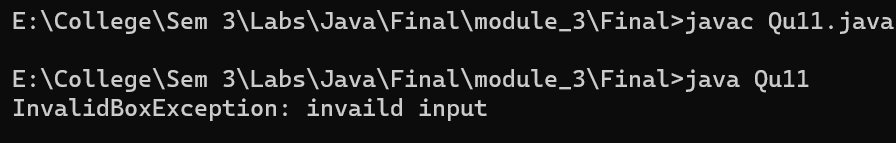
            System.out.println(e);

        }

    }

}

**Output**

****

**Question 12:**

Write a program to create and handle User Defined CheckedException – InsufficientFundsException, generated while withdrawing amount more than available balance.Create necessary class and methods to support this scenario.

**Code**

class InsufficientFunds extends Throwable {

    InsufficientFunds(String s) {

        super(s);

    }

}

class IllegalTransfer extends Throwable {

    IllegalTransfer(String s) {

        super(s);

    }

}

class BankAccount {

    String name;

    int balance;

    BankAccount(String s, int b) {

        name = s;

        balance = b;

    }

    void withdraw(int a) {

        try {

            if (a > balance) {

                throw new InsufficientFunds("insufficient balance in the account");

            } else {

                balance = balance - a;

                System.out.println("Withdraw Done");

            }

        } catch (InsufficientFunds e) {

            System.out.println(e);

        }

    }

    void deposit(int a) {

    }

}

public class Qu12 {

    public static void main(String[] args) {

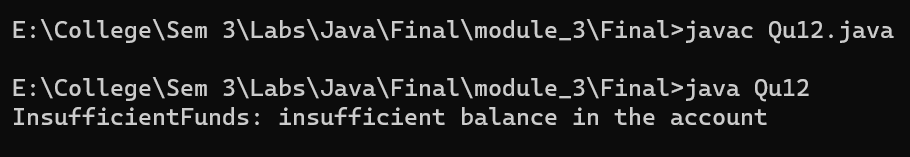
        BankAccount a1 = new BankAccount("abc", 100);

        a1.withdraw(1000);

    }

}

**Output**

****

**Question 13:**

Write a program to create an ArrayList of Products. Traverse the list and print it to the console. Provide a searching of product on name basis using contains() method of List.

**Code**

import java.util.Scanner;

import java.util.ArrayList;

public class Qu13 {

    public static void main(String[] args) {

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

        products.add("samsung");

        products.add("apple");

        products.add("mi");

        products.add("vivo");

        products.add("oneplus");

        System.out.println(products + " ");

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the product to search in ArrayList");

        String name = sc.nextLine();

        if (products.contains(name)) {

            System.out.println("Product found");

        } else {

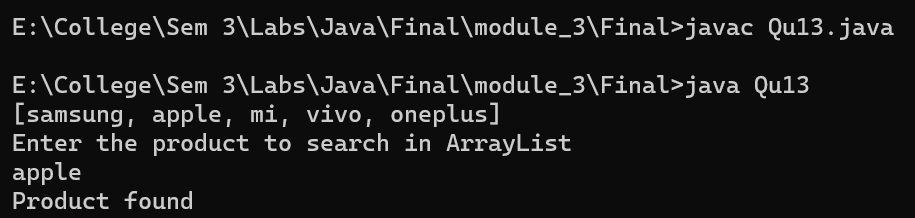
            System.out.println("Product not found");

        }

    }

}

**Output**

****

**Question 14:**

Write a program, to create a TreeSet of Products.Traverse it and provide ordering on base of id.

**Code**

import java.util.\*;

class Product {

    String name;

    int Id;

    Product(int i, String name) {

        this.Id = i;

        this.name = name;

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public int getId() {

        return Id;

    }

}

class idComparator implements Comparator<Product> {

    public int compare(Product p1, Product p2) {

        return p1.getId() - p2.getId();

    }

}

public class Qu14 {

    public static void main(String[] args) {

        TreeSet<Product> set = new TreeSet<Product>(new idComparator());

        set.add(new Product(1000, "samsung"));

        set.add(new Product(02, "apple"));

        set.add(new Product(300, "oneplus"));

        set.add(new Product(4, "vivo"));

        set.add(new Product(555, "mi"));

        for (Product p : set) {

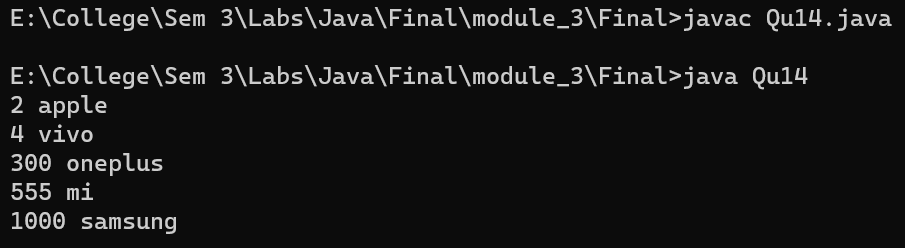
            System.out.println(p.Id + " " + p.name);

        }

    }

}

**Output**

****

**Question 15:**

Write a program to create a TreeSet of Products. Traverse it and provide ordering on basis of name(Use comparator interface).

**Code**

//Write a program to create a TreeSet of Products. Traverse it and provide ordering on basis of name(Use comparator interface).

import java.util.\*;

class Product{

    String name;

    int Id;

    Product(int i,String name){

        this.Id=i;

        this.name=name;

    }

    public String getName(){

        return name;

    }

    public void setName(String name){

        this.name=name;

    }

    public int getId(){

        return Id;

    }

}

class nameComparator implements Comparator<Product>{

    public int compare(Product p1,Product p2){

        return p1.getName().compareTo(p2.getName());

    }

}

public class Qu15 {

    public static void main(String[] args) {

        TreeSet<Product> set = new TreeSet<Product>(new nameComparator());

        set.add(new Product(1000, "samsung"));

        set.add(new Product(02, "apple"));

        set.add(new Product(300, "oneplus"));

        set.add(new Product(4, "vivo"));

        set.add(new Product(555, "mi"));

        for (Product p : set) {

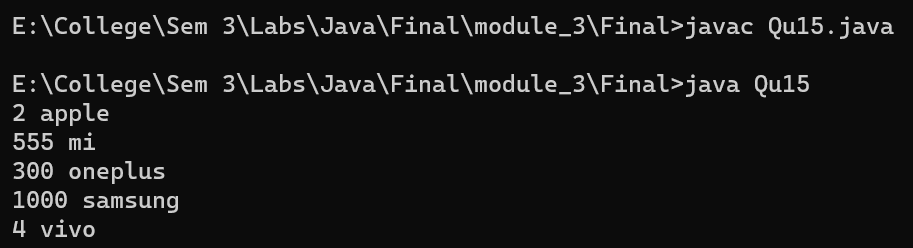
            System.out.println(p.Id + " " + p.name);

        }

    }

}

**Output**

****

**Question 16:**

Write a program to create a HashSet of Products.Demonstrate that no duplicates value are allowed in HashSet.

**Code**

import java.util.\*;

public class Qu16 {

    public static void main(String[] args) {

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

        set.add("samsung");

        set.add("apple");

        set.add("mi");

        set.add("vivo");

        set.add("oneplus");

        set.add("samsung");

        set.add("apple");

        set.add("mi");

        set.add("vivo");

        set.add("oneplus");

        for (String p : set) {

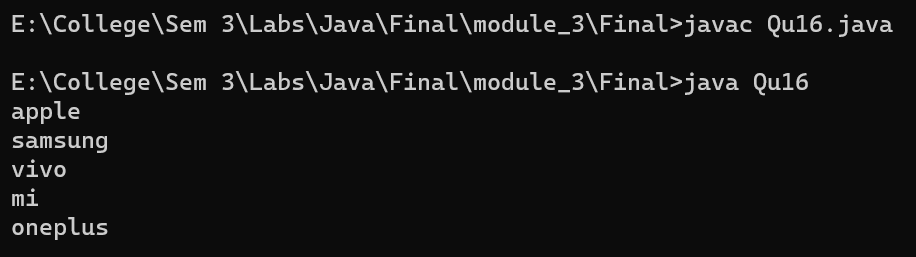
            System.out.println(p);

        }

    }

}

**Output**

****

**Question 17:**

Write a program to demonstrate Thread creation using Runnable interface. While main thread prints 1 to 50 with 1 millisecond pose and child thread print 1 to 100 using 0.5 millisecond pose. Print the name of child and main thread. Main thread needs to wait for child thread to complete.

**Code**

import java.util.\*;

public class Qu17 {

    public static void main(String[] args) {

        Runnable r1 = new Runnable() {

            public void run() {

                for (int i = 1; i <= 50; i++) {

                    System.out.println("Main thread--" + " : " + i);

                    try {

                        Thread.sleep(1000); // main Thread

                    } catch (InterruptedException e) {

                        System.out.println("Thread in sleep mode");

                    }

                }

            }

        };

        for (int i = 1; i <= 100; i++) {

            System.out.println("child thread--" +  " : " + i);

            try {

                Thread.sleep(500); // child Thread

            } catch (InterruptedException e) {

                System.out.println("Thread in sleep mode");

            }

        }

        Thread t1 = new Thread(r1);

        t1.start();

        try {

            t1.join();

        } catch (InterruptedException e) {

            throw new RuntimeException(e);

        }

    }

}