

Java Lab Codes

21 November 2022 11:14

- 0) Reverse a 4 digit number (if number_of digit! =4 reject the input attempt)

```
import java.util.*;
class A1{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        if(n<=999 && n>9999){
            System.out.println("Wrong input"
        );
        }
        else{
            int x = 4;
            int num = 0;
            while(x > 0){
                int rem = n % 10;
                num = (num * 10) + rem;
                n = n/10;
                x--;
            }
            System.out.println(
"Rversed Number = " + num);
        }
    }
}
```

- 1) Find $1 - \frac{1}{2^2} + \frac{1}{3^2} - \frac{1}{4^2} + \dots$ up to number of terms taken from Command line.

```
import java.util.Scanner;
import java.lang.Math;
public class qs1 {
    public static void main(String[] args) {
```

```
Scanner sc=new Scanner(System.in);
System.out.print("Enter the vvalue of n: ");
int n=sc.nextInt();
double sum=0,sign=1;
for (int i = 1; i <= n; i++) {
    sum=sum+sign*(1/Math.pow(i, 2));
    sign=sign*-1;
}
System.out.println(sum);
sc.close();
}
```

- 2) Calculate two maximum nos. of an array; take array element at runtime.

```
import java.util.*;
import java.io.*;
class A3{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for(int i=0;i<n;i++){
            a[i] = sc.nextInt();
        }
        Arrays.sort(a);
        System.out.println(a[n-1]+" "+a[n-2]);
    }
}
```

- 3) Using enhanced for loop, display contents of an integer array (size at least six).

```
import java.util.*;
class A2{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for(int i=0;i<n;i++){
            a[i] = sc.nextInt();
        }
        for(int i : a){
            System.out.print(i+" ");
        }
    }
}
```

- 4) Write a JAVA program to find area and volume of Cone using constructors and Command Line Input.

```
import java.util.*;
import java.io.*;
class A4{
    float r,h;
    A4(float r, float h){
        this.r = r;
        this.h = h;
    }
    void area(){
        float x = (22/7)*r*h + (22/7)*r*r;
        System.out.println(x);
    }
    void volume(){
        float x = (22/7)*r*r*h/3;
        System.out.println(x);
    }
    public static void main(String args[]){
        float a = Float.parseFloat(args[0]);
        float b = Float.parseFloat(args[1]);
        A4 ob = new A4(a, b);
        ob.area();
        ob.volume();
    }
}
```

- 5) Find all prime numbers between 0 & 200.

```
class A5{  
    public static void main(String args[]){  
        for(int i=2;i<=200;i++){  
            int flag = 1;  
            for(int j=2;j<i;j++){  
                if(i%j == 0){  
                    flag = 0;  
                    break;  
                }  
            }  
            if(flag == 1)  
                System.out.print(i+" ");  
        }  
    }  
}
```

6) play first 10 number of Fibonacci series

```
class A6{
    public static void main(String args[]){
        int a = 0;
        int b = 1;
        System.out.print(a+" ");
        System.out.print(b+" ");
        int n = 8;
        while(n > 0){
            int x = a+b;
            System.out.print(x+" ");
            a = b;
            b = x;
            n = n-1;
        }
    }
}
```

- 7) Find $1 - \frac{1}{2^2} + \frac{1}{3^2} - \frac{1}{4^2} \dots \dots$ up to number of terms taken from Command line.

```
class A7{
    public static void main(String args[]){
        int n = Integer.parseInt(args[0]);
        float sum_e = 0, sum_o= 0;
        for(int i=1;i<=n;i++){
            if(i%2 == 1)
                sum_e = sum_e + (1/(float)(i*i));
            else if(i%2 == 0)
                sum_o = sum_o + (1/(float)(i*i));
        }
        System.out.println(sum_e - sum_o);
    }
}
```

- 8) Write a JAVA program to find area and volume of Cone using constructors and Command Line Input.

```
import java.util.*;
class Cone{
    float r, h;
    Cone(float r, float h){
        this.r = r;
        this.h = h;
    }
    void area(){
        float x = ((r*r) + (h*h));
        x = (float)(Math.sqrt(x));
        float y = (22/7)*x*(x+r);
        System.out.println(y);
    }
    void volume(){
        float x = (22/7)*r*r*(h/3);
        System.out.println(x);
    }
}
class A8{
    public static void main(String args[]){
        float a = Float.parseFloat(args[0]);
        float b = Float.parseFloat(args[1]);
        Cone ob = new Cone(a,b);
        ob.area();
        ob.volume();
    }
}
```

9) Calculate GCD of two numbers feed up by Command line.

```
class A9{
    public static void main(String args[]){
        int a = Integer.parseInt(args[0]);
        int b = Integer.parseInt(args[1]);
        int gcd = 1;
        for(int i=1;i<=a && i<=b; i++){
            if((a % i == 0) && (b % i == 0))
                gcd = i;
        }
        System.out.println(gcd);
    }
}
```

- 10) Show that static variable of a class only have one copy for different objects but instance variable may have separate copy for individual object.

```
public class Copy {  
    int x;  
    static int y;  
    Copy(int x,int y){  
        this.x=x;  
        this.y=y;  
    }  
    public static void main(String[] args) {  
        Copy ob1=new Copy(20,30);  
        System.out.println("Object 1 non-Static value : "+ob1.x);  
        System.out.println("Object 1 Static value : "+ob1.y);  
        Copy ob2=new Copy(40,60);  
        System.out.println("Object 2 non-Static value : "+ob2.x);  
        System.out.println("Object 2 Static value : "+ob2.y);  
        System.out.println("Object 1 Static value : "+ob1.y);  
    }  
}
```

- 11) Create a class „Figure“ and make parallelepiped, cube, cylinder and spheres as its object and calculate their surface area only by using method overloading.

```
class Figure{
    void area(int a){
        float area=6*a*a;
        System.out.println("Area of the cube is: "+area);
    }
    void area(int a, int b, int c){
        float x= (a*a)+(b*b)+(c*c);
        float area=(float)Math.sqrt(x);
        System.out.println("Area of the paralelopiped is: "+area);
    }
    void area(float r,float h){
        double area=2*3.14*(r*h +r*r);
        System.out.println("Area of the cylinder is: "+area);
    }
    void area(double r){
        double area=4*3.14*r*r;
        System.out.println("Area of the cube is: "+area);
    }
    public static void main(String args[]){
        Figure ob=new Figure();
        ob.area(2);
        ob.area(12,21,23);
        ob.area(10,20);
        ob.area(5.5);
    }
}
```

- 12) Create a class Parent having instance variables id, name and address. Create a class ChildOne having instance variables id, name, address and marks. Also create another class ChildTwo with instance variables id, name, address, qualification and salary. Design the program and use object of each class from main().

```
class Parent{
    int id = 1;
    String name = "ABCD";
    String address = "Howrah";
}
class ChildOne{
    int id = 2;
    String name = "EFGH";
    String address = "SEALDAH";
    int marks = 100;
}
class ChildTwo{
    int id = 3;
    String name = "IJKL";
    String address = "Kolkata";
    String qualificatioion = "Graduate";
    int salary = 10000;
}
class q12{
    public static void main(String args[]){
        Parent ob1 = new Parent();
        ChildOne ob2 = new ChildOne();
        ChildTwo ob3 = new ChildTwo();
        System.out.println(ob1.name+" "+ob2.marks+" "+ob3.salary);
    }
}
```

- 13) Check if "Tech" presents in "University of Technology" or not. If yes return its position.

```
class q13{
    public static void main(String args[]){
        String s="University of Technology";
        if(s.contains("Tech"))
            System.out.println(s.indexOf("Tech"));
        else
            System.out.println("Not present");
    }
}
```

- 14) Create a class with variable(s) and method(s) under package pOne. Now create a class under package pTwo, which is subclass of firstly created class. In the method here (i.e. class of pTwo) call variable(s) and method(s) of previous class (i.e. class of pOne). Now from Main (under working directory) access second class's members.

```
package P1;
public class A{
    public int x = 17;
    public void show(){
        System.out.println(" In A");
    }
}
```

```
package P2;
import P1.A;
class B extends A{
    void display(){
        System.out.println(x);
        System.out.println("Calling method of class A: ");
        show();
        System.out.println("Method of B");
    }
}
class Test{
    public static void main(String args[]){
        B ob = new B();
        ob.display();
    }
}
```

- 15) Consider the Student class that a student studies three subjects. Each subject has a title, internal-marks and external-marks. Write a Program to define such three students for each of which shows status of individual subject (i.e. fail or pass) and total marks for the subjects.

```
class Student {
    int total;

    void sub1(String title, int internal, int external) {
        total = 0;
        total = total + (internal + external);
        System.out.println("Total marks in " + title + " is " + total);
        if (total < 35) {
            System.out.println("Fail");
        } else {
            System.out.println("Pass");
        }
    }

    void sub2(String title, int internal, int external) {
        total = 0;
        total = total + (internal + external);
        System.out.println("Total marks in " + title + " is " + total);
        if (total < 35) {
            System.out.println("Fail");
        } else {
            System.out.println("Pass");
        }
    }
}
```

```
total = 0;
total = total + (internal + external);
System.out.println("Total marks in " + title + " is " + total);
if (total < 35) {
    System.out.println("Fail");
} else {
    System.out.println("Pass");
}

void sub3(String title, int internal, int external) {
    total = 0;
    total = total + (internal + external);
    System.out.println("Total marks in " + title + " is " + total);

    if (total < 35) {
        System.out.println("Fail");
    } else {
        System.out.println("Pass");
    }
}

class q15 {
    public static void main(String args[]) {
        Student st1 = new Student();
        Student st2 = new Student();
        Student st3 = new Student();

        st1.sub1("History", 23, 70);
        st1.sub2("Geography", 21, 8);
        st1.sub3("Math", 30, 70);

        st1.sub1("History", 24, 67);
        st1.sub2("Geography", 12, 23);
        st1.sub3("Math", 9, 7);

        st1.sub1("History", 17, 56);
        st1.sub2("Geography", 17, 70);
        st1.sub3("Math", 30, 9);
    }
}
```

- 16) Take a string from keyboard and convert into a new character array .

```
import java.util.*;
class q16{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the word");
        String s=sc.nextLine();
        char[] ans=s.toCharArray();
        System.out.print(Arrays.toString(ans));
    }
}
```

- 17) Create an interface containing three methods, in a package “pkgOne”. Implement the interface from a class under package “pkgTwo”. From main, under working directory, create object of the class and call methods of interface.

```
public interface X{
    void A();
    void B();
    void C();
}
```

```
package P2;
import P1.X;
class Y implements X{
    public void A(){
        System.out.println("In A");
    }
    public void B(){
        System.out.println("In B");
    }
    public void C(){
        System.out.println("In C");
    }
    public static void main(String args[]){
        Y ob =new Y();
        ob.A();
        ob.B();
        ob.C();
    }
}
```

- 18) Create a multilevel inheritance of level three and each class have a method void show(). Now from the lowest level class use the show() method of top most class.

```
class A{
    void show(){
        System.out.println("Parent class A");
    }
}
class B extends A{
    void show(){
        System.out.println("Subclass of A");
        A ob=new A();
        ob.show();
    }
}
class C extends B{
    void show(){
        System.out.println("Subclass of B");
        B ob=new B();
        ob.show();
    }
}
class q18{
    public static void main(String args[]){
        A ob1=new A();
        ob1.show();
        B ob2=new B();
        ob2.show();
        C ob3=new C();
        ob3.show();
    }
}
```

- 19) Write a program to take a sentence and convert it into string arrays and sort the words using any Sorting technique.

```
import java.util.*;
class q19{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the sentence");
        String s=sc.nextLine();
        String words[]={s.split(" "});
        for(int i=0;i<words.length-1;i++){
            for(int j=i+1;j<words.length;j++){
                if(words[i].compareTo(words[j]) > 0){
                    String temp=words[i];
                    words[i]=words[j];
                    words[j]=temp;
                }
            }
        }
        System.out.print(Arrays.toString(words));
    }
}
```

20) A class called Television has the following attributes:

- a. Maker
- b. Size of the screen
- c. Date of purchase of the TV
- d. Is it a color TV or black-white TV

Define a class Television. Define a method for displaying the attributes value of a TV. Use them from main().

```
import java.util.*;
class Television{
    void display(){
        Scanner sc=new Scanner(System.in);
        System.out.println("Maker");
        String maker=sc.nextLine();
        System.out.println("Size of the screen");
        String size=sc.nextLine();
        System.out.println("Date of Purchase");
        String date=sc.nextLine();
        System.out.println("Type");
        String type=sc.nextLine();
        System.out.println(maker+ " "+ size+ " "+ date+ " "+ type);
    }
}
class q20{
    public static void main(String args[]){
        Television ob=new Television();
        ob.display();
    }
}
```

21) 1- 2^ 2+3 ^3 -4 ^4+ up to number of terms from keyboard.

```
import java.util.*;
import java.io.*;
class A21{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int sum_e=0, sum_o=0;
        for(int i=1;i<=n;i++){
            if(i%2 == 0){
                sum_e = sum_e + (i*i);
            }
            if(i%2 == 1){
                sum_o = sum_o + (i*i);
            }
        }
        System.out.println(sum_o - sum_e);
    }
}
```

- 22) Write a program in Java to extract a portion of a character string and print the extracted string.
Assume that „m“ characters are extracted, starting with the n-th character.

```
import java.util.*;
import java.io.*;
class A22{
    public static void main(String args[]) throws Exception{
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String s = br.readLine();
        int n = s.length();
        Scanner sc = new Scanner(System.in);
        int m = sc.nextInt();
        int j = sc.nextInt();
        String str = "";
        while(m > 0 && j < n){
            str = str + s.charAt(j);
            j++;
            m--;
        }
        System.out.println(str);
    }
}
```

23) Write a Tree class to represent the trees in a firm. A Tree has the following attributes: Tree Code, height, Base Width. Write a method annualUpdate(). Which updates the height, basewidth, and the amount spent so far on this tree.

② Define a Mango class which is derived class from the Tree class. In addition to the attributes of a Tree, the Mango class has attribute „yield“. Over ride the annualUpdate() method and define a method displayTree() suitably.

③ A garden has two trees(non-mango) and two Mango trees. Define a class called Garden and create these trees and display them.

```
class Tree{  
    String code;  
    int height;  
    int baseW;  
    void annualupdate(int h, int b, int amt){  
        height = h;  
        baseW = b;  
        System.out.println(height+" "+baseW+" "+amt);  
    }  
}  
  
class Mango extends Tree{  
    String yeild = "";  
    int a;  
    void annualupdate(int h, int b, int amt){  
        height = h;  
        baseW = b;  
        a = amt;  
        System.out.println(height+" "+baseW+" "+amt);  
    }  
    void displayTree(String y){  
        yeild = y;  
        System.out.println(height+" "+baseW+" "+a+" "+yeild);  
    }  
}  
class A23{  
    public static void main(String args[]){  
        Tree t1 = new Tree();  
        Tree t2 = new Tree();  
        t1.annualupdate(210,50,5000);  
        t2.annualupdate(280,22,1000);  
        Mango t3 = new Mango();  
        Mango t4 = new Mango();  
        t3.annualupdate(230,56,1111);  
        t4.annualupdate(240,45,2333);  
        t3.displayTree("good");  
        t4.displayTree("bad");  
    }  
}
```

24) Create an interface containing three methods, in a package "pkgOne". Implement the interface from a class under package "pkgTwo". From main, under working directory, create object of the class and call methods of interface.

25) Take a string from keyboard (length <=30) and make a char array (of length 50 but fill first 10 places). Now append the string to that char array.

```
package com.Lot;

import java.util.Scanner;
import java.util.*;
public class Q25 {
    public static void main(String[] args) {
        StringBuilder str=new StringBuilder("Java");
        char [] arr=new char[50];
        Scanner sc=new Scanner(System.in);
        arr[0]='a';
        arr[1]='b';
        arr[2]='c';
        arr[3]='d';
        arr[4]='e';
        arr[5]='f';
        arr[6]='g';
        arr[7]='h';
        arr[8]='i';
        arr[9]='j';
        arr[10]='k';
        str=str.append(arr);
        System.out.println(str);
    }
}
```

26) Design a class to represent a Bank Account. Include the following things:

Fields

- ❑ Name of the depositor
 - ❑ Account number
 - ❑ Type of account
 - ❑ Balance amount in the account
- Methods
- ❑ To assign initial values
 - ❑ To deposit an amount
 - ❑ To withdraw an amount after checking balance

② To display the name and balance

```
package com.Lot;
import java.util.*;
public class Q26 {
    String name;
    int acc;
    String type;
    double balance;
    Q26(String name,int acc,String type,double balance){
        this.name=name;
        this.acc=acc;
        this.type=type;
        this.balance=balance;
    }
    void deposit(double amount) {
        balance=balance+amount;
        System.out.println("Present balance:"+balance);
    }
    void withdrawal(double amount) {
        if(balance>amount) {
            balance=balance-amount;
            System.out.println("Present balance:"+balance);
        }
        else {
            System.out.println("Withdrawal not possible");
        }
    }
    void display() {
        System.out.println("Name:"+name+"Balance:"+balance);
    }
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter your name,account number,account type,balance:");
        String name=sc.next();
        int acc=sc.nextInt();
        String type=sc.next();
        int balance=sc.nextInt();
        Q26 b=new Q26(name,acc,type,balance);
        int ch=sc.nextInt();
        if(ch==1) {
            System.out.println("Enter the amount to deposit:");
            double amount=sc.nextDouble();
            b.deposit(amount);
        }
        else if(ch==2) {
            System.out.println("Enter the amount to withdrawal:");
            double amount=sc.nextDouble();
            b.withdrawal(amount);
        }
        else if(ch==3){
```

```
        double amount=sc.nextDouble();
        b.withdrawal(amount);
    }
    else if(ch==3){
        b.display();
    }
    else
        System.out.println("Invalid choice");
}
}
```

- 27) Check if "Academy" is present in "Academy of Technology" or not. If yes return its position.

```
package com.Lot;

public class Q27 {
    public static void main(String[] args) {
        String s1="Academy of Technology";
        String s2="Academy";
        boolean b=s1.contains(s2);
        if(b==true)
            System.out.println(b+" "+s1.indexOf(s2));
        else
            System.out.println("String not present");
    }
}
```

- 28) Make an abstract class with two concrete methods and no abstract method at all. From your Main class which have main method call the concrete methods.

```
package com.Lot;
abstract class abs{
    void m1() {
        System.out.println("Hi");
    }
    void m2() {
        System.out.println("Hello");
    }
}
public class Q28 extends abs{
    public static void main(String[] args) {
        Q28 q28=new Q28();
        q28.m1();
        q28.m2();
    }
}
```

- 29) Consider the string: "This is a very good practice to use good constructor in good coding of a class, good is good". Now replace „good“ by „bad“.

```
package com.Lot;
public class Q29 {
    public static void main(String[] args) {
        String str="This is a very good practice to use good constructor in good coding of a class, good is good";
        System.out.println(str.replaceAll("good", "bad"));
    }
}
```

- 30) Check whether the letter „a“ or „A“ presents in an input string; find the position and replace the letter by „x“ or „X“ and display the string.

```
import java.util.*;
public class Main
{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the string");
        String s = sc.nextLine();
        int n=s.length();
        char c[] = s.toCharArray();
        char c1='a';
        char c2='A';
        char d1='x';
        char d2='X';

        for(int i=0;i<n;i++)
        {
            if(c[i] == c1)
            {
                c[i] = d1;
                System.out.println("a is present in at index " + i);
            }
            if(c[i] == c2)
            {
                c[i] = d2;
                System.out.println("A is present at index " + i );
            }
        }

        for(int i=0;i<n;i++)
        {
            System.out.print(c[i]);
        }
    }
}
```

- 31) Create a class, make method inside it which return an object and also take object as parameter of this method (The returned object and parameterized object should not from same class, is preferable)

```
package com.Lot;
class Test{

}

public class Q31 {
    public Test z(Test o) {
        Test t1=new Test();
        return t1;
    }
    public static void main(String[] args) {
        Q31 d2=new Q31();
        Test t=new Test();
        System.out.println("Hi:");
        d2.z(t);
    }
}
```

- 32) Show that access specifier for overridden method (in child) must be broaden (i.e. should not be narrower) than access specifier of original method of parent.

```
package com.Lot;
class Parent{
    public void m1() {
        System.out.println("Yo");
    }
}
public class Q32 extends Parent{
    /*scope of access modifiers can't be reduced while overriding so this code will result to C.E.
     *void m1() {
     System.out.println("Meu");
    }*/
    public void m1() {
        System.out.println("Hello");
    }
    public static void main(String[] args) {
        Parent p=new Q32();
        p.m1();
    }
}
```

33) Convert the following sentence into array of Strings and sort it:

'This is a very good practice to use good constructor in coding of a good class, good is always good"

```
package com.Lot;

import java.util.Arrays;

public class Q33 {
    public static void main(String[] args) {
        String str="This is a very good practice to use good constructor in coding of a good class, good is always good";
        String []str1=str.split(" ");
        for(int i=0;i<str1.length;i++)
            System.out.println(str1[i]);
        Arrays.sort(str1);
        System.out.println(Arrays.toString(str1));
    }
}
```

- 34) Create a class EMP having instance variable name and id. Create its subclass (say Scientist) which has instance variable no_of_publication and experience. Now create its subclass, say Dscientist which has instance variable award. Put a method: public String toString(){ } in every class where you describe about the class and from main create object of each class and print each object.

```
package Assignment_03; //comment before run

class EMP{
    int id;
```

```
package Assignment_03; //comment before run

class EMP{
    int id;
    String name;
    EMP(int id, String name){
        this.id=id;
        this.name=name;
    }
    public String toString(){
        System.out.println(id);
        System.out.println(name);
        return "In EMP class";
    }
}

class Scientist extends EMP{
    int no_of_publication,experience;
    Scientist(int id, String name, int no_of_publication, int experience){
        super(id, name);
        this.no_of_publication=no_of_publication;
        this.experience=experience;
    }
    public String toString(){
        System.out.println(no_of_publication);
        System.out.println(experience);
        return "In Scientist class";
    }
}

class DScientist extends Scientist{
    int award;
    DScientist(int id, String name, int no_of_publication, int experience, int award){
        super(id, name, no_of_publication, experience);
        this.award=award;
    }
    public String toString(){
        System.out.println(award);
        return "In DScientist class";
    }
}

public class qs4 {
    public static void main(String[] args) {
        int id=12,no_of_publication=29,experience=15,award=52;
        String name="SAYAN";
        EMP e=new EMP(id, name);
        Scientist s=new Scientist(id, name, no_of_publication, experience);
        DScientist d=new DScientist(id, name, no_of_publication, experience, award);
        e.toString();
        s.toString();
        d.toString();
    }
}
```

```
        a.toString(),  
    }  
}
```

- 35) Show that static block is executed at the time of class loading

```
class A35{  
    A35(){  
        System.out.println("Constructor");  
    }  
    {  
        System.out.println("Instance");  
    }  
    static{  
        System.out.println("Static A");  
    }  
    public static void main(String args[]){  
        new A35();  
    }  
}
```

- 36) Write a program in Java which will read a string and rewrite it in the alphabetical order. For example, the word STRING should be written as GINRST.

```
import java.util.*;
public class Main
{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the string");
        String s = sc.nextLine();
        int n=s.length();
        char c[] = s.toCharArray();
        for(int i=0;i<n;i++)
        {
            for(int j=0;j<n-1-i;j++)
            {
                int a=(int)c[j];
                int b=(int)c[j+1];
                if(a>b)
                {
                    char m = c[j];
                    c[j] = c[j+1];
                    c[j+1] = m;
                }
            }
            for(int i=0;i<n;i++)
            {
                System.out.print(c[i]);
            }
        }
    }
}
```

- 37) Create an abstract class TwoD with fields double sideOne and double sideTwo. The class has with suitable constructor and an abstract method for calculating area. Now make Rectangle and Triangle as subclasses of above and override the method and call it from main() for each of subclasses.

```
abstract class TwoD
{
    double s1=0.0;
    double s2=0.0;
    public TwoD()
    {
        this.s1=s1;
        this.s2=s2;
    }
    abstract void area(double s1,double s2);
}

class Rect extends TwoD
{
    void area(double a,double b)
    {
        double c = a*b;
        System.out.println("The area of the rectangle is : "+c);
    }
}

class Tri extends TwoD
{
    void area(double a,double b)
    {
        double c = 0.5*a*b;
        System.out.println("The area of the triangle is : "+c);
    }
}

public class Main
{
    public static void main(String args[])
    {
        Rect ob = new Rect();
        ob.area(4.0,3.0);
        Tri obl = new Tri();
        obl.area(4.0,3.0);
    }
}
```

- 38) Find length of a string taken from keyboard and also find the length of that string except front and end spaces.

```
class A38{
    public static void main(String args[]){
        String s = "      Rishabh      ";
        int ans = 0, p =0;
        System.out.println("Length of string = : "+ s.length());
        for(int i=0;i<s.length();i++){
            char q = s.charAt(i);
            char a = ' ';
            if(q == a){
                p++;
            }
            else{
                break;
            }
        }
        for(int i=s.length()-1;i>=0;i--){
            char q = s.charAt(i);
            char a = ' ';
            if(q == a){
                p++;
            }
            else{
                break;
            }
        }
        ans = s.length();
        System.out.println("Length of string without wight spaces= : "+ (ans-p));
    }
}
```

39) Write a java program to find: $1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} \dots \dots$ up to n terms.

```
import java.util.*;
public class Main
{
    static int fact(int k)
    {
        if(k<=1)
            return 1;
        else
            return k*fact(k-1);
    }

    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the no. of terms");
        int n = sc.nextInt();
        double sum =0.0;
        for(int i=0;i<n;i++)
        {
            if(i%2==0)
            {
                sum+=1.0/fact(i);
            }
            else
            {
                sum-=1.0/fact(i);
            }
        }
        System.out.println("The sum of the series is : "+sum);
    }
}
```

- 40) Generate password from initials of one's first_name, middle_name, last_name and with last four digit of your roll_no (if middle name is not present, it won't come).

```
import java.util.*;
public class Main
{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the string");
        String s = sc.nextLine();
        System.out.println("Enter the roll no.");
        int r = sc.nextInt();
        int k=4;int sum=0;int w=0;
        String str="";
        while(k>0)
        {
            int a=r%10;
            sum=a+(sum*10);
            r=r/10;
            k--;
        }
        int l=4;
        while(l>0)
        {
            int d=sum%10;
            w=d+(w*10);
            sum=sum/10;
            l--;
        }
        String b=Integer.toString(w);
        int n=s.length();
        char c[] = s.toCharArray();
        str=str+c[0];
        for(int i=0;i<n;i++)
        {
            if(c[i] == ' ')
            {
                str=str+c[i+1];
            }
        }
        str=str+b;
        System.out.println(str);
    }
}
```

41) Write a java program to show that parent block is executed before child object is created.

```
class A extends Main
{
    void display()
    {
        System.out.println("Child class");
    }
}

public class Main
{
    static{
        System.out.println("Parent block");
    }

    public static void main(String args[])
    {
        A ob = new A();
        ob.display();
    }
}
```

42) Define an object reference and initialize it to null. Try to call a method through this reference.

Now wrap the code in a try-catch clause to catch the exception

```
package com.Aot.test;

public class Exception4 {
    void m1()
    {
        System.out.println("Hi");
    }
    public static void main(String[] args) {
        try {
            Exception4 e=new Exception4();
            e=null;
            e.m1();
        }
        catch(NullPointerException npe){
            System.out.println(npe);
        }
    }
}
```

- 43) Write a program to create a user defined exception named PayOutOfBoundsException (provided the monthly salary of a person is less than Rs. 10,000/-) and fire the exception.

```
package com.Aot.test;
import java.util.*;
class PayOutOfBoundsException extends Exception{

}

public class Exception6 {

    public static void main(String[] args) {
        System.out.println("Enter Salary:");
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        try {
            if(n<10000)
                throw new PayOutOfBoundsException();
            else
                System.out.println("Hello");
        }
        catch(PayOutOfBoundsException poobe) {
            System.out.println("Exception Handled");
        }
    }
}
```

21 November 2022 14:59