

1. Write a program to find the distance between two points.

Class diagram:

Distance

~x1: int

~x2: int

~y1: int

~y2: int

~dis: double

```
import java.io.*;
import java.util.*;
class Distance
{
    public static void main(String args[])
    {
        Scanner in=new Scanner(System.in);
        int x1,y1,x2,y2;
        double dis;
        System.out.println("ENTER THE COORDINATES
OF FIRST POINT:");
        x1=in.nextInt();
        y1=in.nextInt();
        System.out.println("ENTER THE COORDINATES
OF SECOND POINT:");
        x2=in.nextInt();
        y2=in.nextInt();
        dis=Math.sqrt(((x2-x1)*(x2-x1))+((y2-
y1)*(y2-y1)));
        System.out.println("DISTANCE:"+dis);
    }
}
```

output:

ENTER THE COORDINATES OF FIRST POINT:

10

20

ENTER THE COORDINATES OF SECOND POINT:

30

40

DISTANCE:28.284271247461902

2. Write a program to find the sum, difference, product, quotient and remainder of two numbers passed as command line argument.

Class diagram :

Numbers

~a:int

~b:int

```
import java.util.Scanner;

public class Numbers {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the value of 'a':");
        int a = scanner.nextInt();

        System.out.print("Enter the value of 'b':");
        int b = scanner.nextInt();

        System.out.println("Sum: " + (a + b));
        System.out.println("Difference: " + (a - b));
        System.out.println("Product: " + (a * b));

        if (b != 0) {
            System.out.println("Quotient: " + (a / b));
            System.out.println("Remainder: " + (a % b));
        } else {
            System.out.println("Cannot divide by zero.");
        }
    }
}
```

```

        scanner.close();
    }
}

```

output:

Enter the value of 'a': 10

Enter the value of 'b': 20

Sum: 30

Difference: -10

Product: 200

Quotient: 0

Remainder: 10

3. Write java program to display Fibonacci series up to a limit.

Class diagram:

Fib

~n1: int=0

~n2: int=1

~n3: int

~limit: int

~i: int

```

import java.io.*;
import java.util.*;
class Fib
{
    public static void main(String args[])
    {
        Scanner in=new Scanner(System.in);
        int n1=0,n2=1,n3,limit,i;
        System.out.println("ENTER THE
LIMIT:");
        limit=in.nextInt();
        System.out.print(n1+"\t"+n2);
        for(i=2;i<limit;i++)
        {
            n3=n1+n2;

```

```

        n1=n2;
        n2=n3;
        System.out.print("\t"+n3);
    }
}

```

output:

ENTER THE LIMIT:

5

0 1 1 2 3

4. Write java program to display Armstrong numbers within a range.

Class diagram:

Armstrong

~l: int

~h: int

~i: int

~rem: int

~sum: int

~num= int

```

import java.io.*;
import java.util.*;
class Armstrong
{
    public static void main(String args[])
    {
        Scanner in=new Scanner(System.in);
        int l,h,i,rem,sum,num;
        System.out.println("ENTER THE LOWER LIMIT
AND UPPER LIMIT:");
        l=in.nextInt();
        h=in.nextInt();
        System.out.println("ARMSTRONG NUMBERS:");
        for(i=l;i<=h;i++)

```

```

        {
            num=i;
            sum=0;
            while (num!=0)
            {
                rem=num%10;
                sum=sum+(rem*rem*rem);
                num=num/10;
            }
            if (sum==i)
                System.out.println(i);
        }
    }
}

```

output:

ENTER THE LOWER LIMIT AND UPPER LIMIT:

100

500

ARMSTRONG NUMBERS:

153

370

371

407

5. Given the sides of a triangle, write a program to check whether the triangle is equilateral, isosceles or scalene and find its area.

Class diagram:

Triangle

~a: int

~b: int

~c: int

~s: double

~area: double

```
import java.io.*;
import java.util.*;
class Triangle
{
    public static void main(String args[])
    {
        int a,b,c;
        double s,area;
        Scanner in=new Scanner(System.in);
        System.out.println("ENTER THE SIDES OF A
TRIANGLE:");
        a=in.nextInt();
        b=in.nextInt();
        c=in.nextInt();
        if(a==b&&b==c)
            System.out.println("EQUILATERAL
TRIANGLE");
        else if(a==b||b==c||a==c)
            System.out.println("ISOSCELESS
TRIANGLE");
        else
            System.out.println("SCALENE
TRIANGLE");
        s=(a+b+c)/2;
        area=Math.sqrt(s*(s-a)*(s-b)*(s-c));
        System.out.println("AREA:"+area);
    }
}
```

output:

ENTER THE SIDES OF A TRIANGLE:

20

20

10

ISOSCELESS TRIANGLE

AREA:96.82458365518542

6. Read an array of 10 or more numbers and write a program to find the

- a) Smallest element in the array
- b) Largest element in the array
- c) Second largest element in the array

class diagram:

Numbers

~n: int

~i: int

~j: int

~temp: int

~a[]: int

```
import java.io.*;
import java.util.*;

class Numbers {
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        int n, i, j, temp;
        int a[] = new int[10];
        System.out.println("ENTER THE NUMBER OF
ELEMENTS:");
        n = in.nextInt();
        System.out.println("ENTER THE ELEMENTS:");
        for (i = 0; i < n; i++) {
            a[i] = in.nextInt();
        }
        for (i = 0; i < n; i++) {
            for (j = 0; j < n - i - 1; j++) {
                if (a[j] > a[j + 1]) {
                    temp = a[j];
                    a[j] = a[j + 1];
                    a[j + 1] = temp;
                }
            }
        }
    }
}
```

```

    }
    System.out.println("SMALLEST ELEMENT:" +
a[0] +
                        "\n LARGEST ELEMENT:" + a[n - 1] +
                        "\n SECOND LARGEST ELEMENT:" + a[n
- 2] ) ;
    }
}

```

output:

ENTER THE NUMBER OF ELEMENTS:

5

ENTER THE ELEMENTS:

2

4

1

6

3

SMALLEST ELEMENT:1

LARGEST ELEMENT:6

SECOND LARGEST ELEMENT:4

7. Write a program to perform base conversion

a) Integer to binary

b) Integer to Octal

c) Integer to Hexadecimal

class diagram:

Base

~num: int

~rem:int

~base: int

~str: String

~dig[]: char

```
import java.io.*;
import java.util.*;
class Base
{
    public static void main(String args[])
    {
        Scanner in =new Scanner(System.in);
        int num,rem,base;
        String str="";
        char
dig[]={ '0','1','2','3','4','5','6','7','8','9','A'
,'B','C','D','E','F'};
        System.out.println("ENTER THE NUMBER:");
        num=in.nextInt();
        System.out.println("ENTER THE BASE TO
CONVERT:");
        base=in.nextInt();
        while (num>0)
        {
            rem=num%base;
            str=dig[rem]+str;
            num=num/base;
        }
        System.out.println(str);
    }
}
```

output:

ENTER THE NUMBER:

10

ENTER THE BASE TO CONVERT:

2

1010

8. Write a program to merge two arrays.

Class diagram:

Merge

~m: int

~n: int

~i: int

~i: int

~j: int

~k: int

~a1[]: int

~a2[]: int

~a3[]: int

```
import java.io.*;
import java.util.*;
class Merge
{
    public static void main(String args[])
    {
        Scanner in=new Scanner(System.in);
        int m,n,i,j,k=0;
        int a1[]=new int[10];
        int a2[]=new int[10];
        int a3[]=new int[20];
        System.out.println("enter the size of
array 1:");
        m=in.nextInt();
        System.out.println("enter the elements:");
        for(i=0;i<m;i++)
        {
            a1[i]=in.nextInt();
        }
        System.out.println("enter the size of
array 2:");
        n=in.nextInt();
```

```
System.out.println("enter the elements:");
for(i=0;i<n;i++)
{
    a2[i]=in.nextInt();
}
i=0;
j=0;
k=0;
while(i<m&& j<n)
{
    if(a1[i]<a2[j])
    {
        a3[k]=a1[i];
        i++;
    }
    else
    {
        a3[k]=a2[j];
        j++;
    }
    k++;
}
if(i>=m)
{
    while(j<n)
    {
        a3[k]=a2[j];
        j++;
        k++;
    }
}
if(j>=n)
{
    while(i<m)
    {
        a3[k]=a1[i];
        i++;
        k++;
    }
}
System.out.println("after merging:");
```

```
        for(i=0;i<m+n;i++)
        {
            System.out.println(a3[i]);
        }
    }
}
```

output:

enter the size of array 1:

3

enter the elements:

10

30

50

enter the size of array 2:

2

enter the elements:

20

40

after merging:

10

20

30

40

50

9. Java Programming Code to Find HCF LCM of Two Numbers

Class diagram:

HcfLcm

~a: int
~b: int
~x: int
~y: int
~t: int
~hcf: int
~lcm: int

```
import java.io.*;
import java.util.*;
class HcfLcm
{
    public static void main(String args[])
    {
        int a, b, x, y, t, hcf, lcm;
        Scanner in = new Scanner(System.in);
        System.out.print("Enter Two Number : ");
        x = in.nextInt();
        y = in.nextInt();
        a = x;
        b = y;
        while(b != 0)
        {
            t = b;
            b = a%b;
            a = t;
        }
        hcf = a;
        lcm = (x*y)/hcf;
        System.out.print("HCF = " +hcf);
        System.out.print("\nLCM = " +lcm);
    }
}
```

output:

Enter Two Number : 16

20

HCF = 4

LCM = 80

10. Write a Java Program to convert centimeter to inch, meter and kilometer

Class diagram:

cmconvert

~cm: double

~inch: double

~mtr: double

~km: double

```
import java.io.*;
import java.util.*;
class cmconvert
{
    public static void main(String args[])
    {
        Scanner in=new Scanner(System.in);
        double cm,inch,mtr,km;
        System.out.println("ENTER THE CENTEMETER
VALUE:");
        cm= in.nextDouble();
        inch=0.3937*cm;
        mtr=0.01*cm;
        km=0.00001*cm;
        System.out.println(cm+"cm is equal to
"+inch+" inches");
        System.out.println(cm+"cm is equal to
"+mtr+" mtr");
        System.out.println(cm+"cm is equal to
"+km+" km");
    }
}
```

output:

ENTER THE CENTEMETER VALUE:

20

20.0cm is equal to 7.874 inches

20.0cm is equal to 0.2 mtr

20.0cm is equal to 2.0E-4 km

11. Write a program to find the trace and transpose of a matrix.

Class diagram:

Matrix

~n: int

~m: int

~i: int

~j: int

~trace: int=0

~a[][]: int

~b[][]: int

```
import java.io.*;
import java.util.*;
class Matrix
{
    public static void main(String args[])
    {
        Scanner in=new Scanner(System.in);
        int n,m,i,j,trace=0;
        int a[][]=new int[10][10];
        int b[][]=new int[10][10];
        System.out.println("ENTER THE ORDER OF THE
MATRIX:");
        m=in.nextInt();
        n=in.nextInt();
        System.out.println("ENTER THE ELEMENTS:");
        for(i=0;i<m;i++)
        {
```

```

        for(j=0;j<n;j++)
        {
            a[i][j]=in.nextInt();
        }
    }
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            b[j][i]=a[i][j];
            if(i==j)
            {
                trace=trace+a[i][j];
            }
        }
    }
    System.out.println("TRACE:"+trace);
    System.out.println("TRANSPOSE");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            System.out.print(b[i][j]);
        }
        System.out.println(" ");
    }
}

```

output:

ENTER THE ORDER OF THE MATRIX:

3

3

ENTER THE ELEMENTS:

1 2 3

4 5 6

7 8 9

TRACE:15

TRANSPOSE

147

258

369

12. Write java program to find the sum of the digits and reverse of a given

number using class and objects

class diagram:

Ten

~num: int

~rem: int

~sum: int=0

~rev: int=0

~sumrev()

```
import java.io.*;
import java.util.*;
class Ten
{
    int num,rem,sum=0,rev=0;
    public static void main(String args[])
    {
        Ten obj=new Ten();
        obj.sumrev();
    }
    void sumrev()
    {
        Scanner in=new Scanner(System.in);
        System.out.println("ENTER THE NUMBER:");
        num=in.nextInt();
        while(num!=0)
        {
            rem=num%10;
            sum=sum+rem;
            rev=rem+(rev*10);
        }
    }
}
```

```

        num=num/10;
    }
    System.out.println("SUM:"+sum);
    System.out.println("REVERSE:"+rev);
}
}

```

output:

ENTER THE NUMBER:

123

SUM:6

REVERSE:321

13. Write a Java Programming Code to Check given string Anagram or Not. If the two strings are anagram to each other, then one string can be rearranged to form the other string. For Example: abc and cba are anagram.

Class diagram:

Anagram

~str1: String

~str2: String

~len1: int

~len2:int

~st1: char[]

~st2:char[]

```

import java.io.*;
import java.util.*;
class Anagram
{
    public static void main(String args[])
    {
        Scanner in=new Scanner(System.in);
        String str1,str2;
        int len1,len2;
    }
}

```

```

        System.out.println("ENTER THE FIRST
STRING:");
        str1=in.next();
        System.out.println("ENTER THE SECOND
STRING:");
        str2=in.next();
        char[] st1=str1.toCharArray();
        char[] st2=str2.toCharArray();
        Arrays.sort(st1);
        Arrays.sort(st2);
        len1=str1.length();
        len2=str2.length();
        if(len1==len2)
        {
            if(Arrays.equals(st1,st2))
                System.out.println("STRINGS ARE
ANAGRAM");
            else
                System.out.println("STRINGS ARE
NOT ANAGRAM");
        }
        else
        {
            System.out.println("DIFFERENT
LENGTH... NOT ANAGRAM");
        }
    }
}

```

output:

ENTER THE FIRST STRING:

silent

ENTER THE SECOND STRING:

listen

STRINGS ARE ANAGRAM

14. Write a Java Program to remove all vowels from a string

Class diagram:

vowel

~str: String

```
import java.io.*;
import java.util.*;
class vowel
{
    public static void main(String args[])
    {
        Scanner in=new Scanner(System.in);
        String str;
        System.out.println("ENTER THE STRING:");
        str=in.nextLine();
        str=str.replaceAll("[aeiouAEIOU]", "");
        System.out.println(str);
    }
}
```

output:

ENTER THE STRING:

welcome

wlcm

15.Using class and objects, write a java program to find the sum of two complex numbers (Hint: Use object as parameter to function).

Class diagram:

Complex

~real: int

~image: int

add(complex c1, complex c2):Complex

```
import java.io.*;
import java.util.*;
class Complex
{
```

```

int real,image;
public static void main(String args[])
{
    Scanner in=new Scanner(System.in);
    Complex c1=new Complex();
    Complex c2=new Complex();
    System.out.println("ENTER THE FIRST
COMPLEX NUMBER:");
    c1.real=in.nextInt();
    c1.image=in.nextInt();
    System.out.println("ENTER THE FIRST
COMPLEX NUMBER:");
    c2.real=in.nextInt();
    c2.image=in.nextInt();
    c1.add(c1,c2);
}
void add(Complex c1,Complex c2)
{
    Complex c3=new Complex();
    c3.real=c1.real+c2.real;
    c3.image=c1.image+c2.image;
    System.out.println("SUM:"+c3.real+"+"+c3.image);
}
}

```

output:

ENTER THE FIRST COMPLEX NUMBER:

10

20

ENTER THE FIRST COMPLEX NUMBER:

30

40

SUM:40+60

16. Write a program to count and display total number of objects created to a class

```

import java.io.*;
import java.util.*;
class NumObj
{
    static int count=0;
    NumObj ()
    {
        count++;
    }
    public static void main(String args[])
    {
        NumObj obj1=new NumObj ();
        NumObj obj2=new NumObj ();
        NumObj obj3=new NumObj ();
        System.out.println("NUMBER OF OBJECTS
CREATED:"+count);
    }
}

```

output:

NUMBER OF OBJECTS CREATED:3

17. Write a java program to find the volume of cube, rectangular box, cylinder using function overloading.

Class diagram:

Seventeen

~length: double

~width:double

~height: double

~radius: double

~area:double

~volume(double l)

~volume(double l, double w, double h)

~volume(double r, double h)

```

import java.io.*;
import java.util.*;
class seventeen
{
    public static void main(String args[])
    {
        seventeen obj=new seventeen();
        Scanner in=new Scanner(System.in);
        double length,width,height,radius,area;
        System.out.println("ENTER THE LENGTH OF
CUBE");
        length=in.nextDouble();
        obj.volume(length);
        System.out.println("ENTER THE RADIUS AND
HEIGHT OF CYLINDER");
        radius=in.nextDouble();
        height=in.nextDouble();
        obj.volume(length,height);
        System.out.println("ENTER THE LENGTH,
WIDTH, HEIGHT OF RECTANGULAR BOX");

        length=in.nextDouble();
        width=in.nextDouble();
        height=in.nextDouble();
        obj.volume(length,width,height);
    }
    void volume(double l)
    {
        System.out.println("VOLUME:"+l*l*l);
    }
    void volume(double l,double w,double h)
    {
        System.out.println("VOLUME:"+l*w*h);
    }
    void volume(double r,double h)
    {
        System.out.println("VOLUME:"+3.14*r*r*h);
    }
}

```

output:

ENTER THE LENGTH OF CUBE

10

VOLUME:1000.0

ENTER THE RADIUS AND HEIGHT OF CYLINDER

10

5

VOLUME:1570.0

ENTER THE LENGTH, WIDTH, HEIGHT OF RECTANGULAR BOX

10

5

2

VOLUME:100.0

18.Create an interface volume with member variable pi and methods readdata () and dispvolume (). Create two classes sphere and cylinder to implement this interface. Write a Java program to find the volume.

Class diagram:

<<interface>>

Volume

~pi:double=3.14;

+readdata():void

+dispvolume():void

Sphere

~r:double

~vol:double

+readdata():void

+dispvolume():void

Cylinder

~r:double;

~h:double

~vol:double

+readdata():void

+dispvolume():void

```
import java.io.*;
import java.util.*;
interface volume
{
    static final double pi=3.14;
    public abstract void readdata();
    public abstract void dispvolume();
}
class sphere implements volume
{
    Scanner in=new Scanner(System.in);
    double r,vol;
    public void readdata()
    {
        System.out.println("ENTER THE RADIUS
SPHERE:");
        r=in.nextDouble();
    }
    public void dispvolume()
    {
        vol=(4/3)*pi*r*r*r;
        System.out.println("VOLUME OF
SPHERE:"+vol);
    }
}
class cylinder implements volume
{
    Scanner in=new Scanner(System.in);
    double r,vol,h;
    public void readdata()
    {
        System.out.println("ENTER THE RADIUS OF
CYLINDER:");
        r=in.nextDouble();
```

```

        System.out.println("ENTER THE HEIGHT OF
CYLINDER:");
        h=in.nextDouble();
    }
    public void dispvolume()
    {
        vol=pi*r*r*h;
        System.out.println("VOLUME OF
CYLINDER:"+vol);
    }
}
class Eighteen
{
    public static void main(String args[])
    {
        cylinder cyl=new cylinder();
        sphere sp=new sphere();
        cyl.readdata();
        cyl.dispvolume();
        sp.readdata();
        sp.dispvolume();
    }
}

```

output:

ENTER THE RADIUS OF CYLINDER:

10

ENTER THE HEIGHT OF CYLINDER:

2

VOLUME OF CYLINDER:628.0

ENTER THE RADIUS SPHERE:

5

VOLUME OF SPHERE:392.5

19. Write a multi thread java program for displaying odd numbers and even numbers up to a limit (Hint: Implement thread using Runnable interface).

Class diagram:

Odd

~i:int

~n:int=10

+run: void

Even

~i:int

~n:int=10

+run: void

OddEvenThread

ob1:Odd

ob2:Even

```
import java.io.*;
import java.util.*;
class Odd implements Runnable
{
    int i,n=10;
    public void run()
    {
        for(i=1;i<n;i=i+2)
        {
            System.out.println("ODD:"+i);
        }
    }
}
class Even implements Runnable
{
    int i,n=10;
    public void run()
    {
        for(i=0;i<n;i=i+2)
        {
            System.out.println("EVEN:"+i);
        }
    }
}
```

```
class OddEvenThread
{
    public static void main(String args[])
    {
        Even ob2=new Even();
        Thread obj2=new Thread(ob2);
        obj2.start();
        Odd ob1=new Odd();
        Thread obj1=new Thread(ob1);
        obj1.start();
    }
}
```

output:

ODD:1

ODD:3

ODD:5

ODD:7

ODD:9

EVEN:0

EVEN:2

EVEN:4

EVEN:6

EVEN:8