

JAVA LAB PROGRAMS

1) Write a java program to display the maximum and minimum of three numbers.

```
import java.util.Scanner;
public class MinMax
{
    public static void main(String [] args)
    {
        Scanner sc = new Scanner(System.in);
        int a,b,c,min,max;
        System.out.print("Enter 3 Nos : ");
        a = sc.nextInt();
        b = sc.nextInt();
        c = sc.nextInt();
        min = max = a;
        if(b>max)
            max = b;
        if(c>max)
            max = c;
        if(b<min)
            min = b;
        if(c<min)
            min = c;
        System.out.println("Max : "+max);
        System.out.println("Min : "+min);
    }
}
```

OUTPUT:

Enter 3 Nos : 5 10 -8

Max : 10

Min : -8

2) Write a java program to find out the factorial of a number.

```
import java.util.Scanner;
public class Factorial
{
    public static void main(String [] args)
    {
        Scanner sc = new Scanner(System.in);
        int n,ans=1,temp;
        System.out.print("Enter a No : ");
        n = sc.nextInt();
        temp = n;
        while(n>0)
        {
            ans = ans * n;
            n--;
        }
        System.out.println(temp+"! = "+ans);
    }
}
```

OUTPUT:

Enter a No : 6

6! = 720

3) Write a java program to find the first 10 numbers in the fibonacci series.

```
public class Fibonacci
{
    public static void main(String [] args)
    {
        int a=0,b=1,c,n=10;
        while(n>0)
        {
            c = a+b;
            System.out.print(a+"\t");
            a=b;
            b=c;
            n--;
        }
    }
}
```

OUTPUT:

0	1	1	2	3	5	8	13
21	34						

4) Write a java program to check whether a number is an armstrong or not.

```
import java.util.Scanner;
import java.lang.Math;
public class Armstrong
```

```

{
    public static void main(String [] args)
    {
        Scanner sc = new Scanner(System.in);
        int n,digits=0,d,temp1,temp2,sum=0;
        System.out.print("Enter a Number: ");
        n = sc.nextInt();
        temp1=temp2=n;
        //To find out Number of Digits in the given Number
        while(n>0)
        {
            digits++;
            n = n/10;
        }
        //To calculate Armstrong 317 = (3^3)+(1^3)+(7^3)
        while(temp1>0)
        {
            d = temp1%10;
            sum+=Math.pow(d,digits);
            temp1 = temp1/10;
        }
        if(temp2 == sum)
            System.out.println(temp2+" is an Armstrong no.");
        else
            System.out.println(temp2+" is not an Armstrong
no.");
    }
}

```

OUTPUT:

Enter a Number: 1634

1634 is an Armstrong no.

5) Write a java program to check whether a number is prime or not.

```
import java.util.Scanner;
import java.lang.Math;
public class Prime
{
    public static void main(String [] args)
    {
        Scanner sc = new Scanner(System.in);
        int n,count=0,i;
        System.out.print("Enter a No : ");
        n = sc.nextInt();
        for(i=1;i<Math.sqrt(n);i++)
        {
            if(n%i == 0)
            {
                count++;
            }
        }
        if(count==1)
            System.out.println(n+" is a Prime Number");
        else
            System.out.println(n+" is not a Prime Number");
    }
}
```

OUTPUT:

Enter a No : 79

79 is a Prime Number

6) Write a java program to display prime numbers within the given range of numbers.

```
import java.util.Scanner;
public class PrimeBtwNos
{
    public static void main(String [] args)
    {
        Scanner sc = new Scanner(System.in);
        int ul,ll,i=0,j,c;
        System.out.print("Enter Lower Limit : ");
        ll = sc.nextInt();
        System.out.print("Enter Upper Limit : ");
        ul = sc.nextInt();
        System.out.println("Prime nos. between "+ll+"-"+ul+"
are: ");
        for(i=ll;i<=ul;i++)
        {
            c=0;
            for(j=1;j<=i/2;j++)
            {
                if(i%j == 0)
                {
                    c++;
                }
            }
            if(c==1)
                System.out.print(i+"\t");
        }
    }
}
```

OUTPUT:

Enter Lower Limit : 11

Enter Upper Limit : 27

Prime nos. between 11-27 are:

11 13 17 19 23

7) Write a java program to check whether a number is a palindrome number.

```
import java.util.Scanner;
public class Palindrome
{
    public static void main(String [] args)
    {
        Scanner sc = new Scanner(System.in);
        int n,temp,d,rev=0;
        System.out.print("Enter a No : ");
        n = sc.nextInt();
        temp = n;
        while(n>0)
        {
            d = n%10;
            rev = (rev*10)+d;
            n = n/10;
        }
        if(rev == temp)
            System.out.println(temp+" is a Palindrome no.");
        else
```

```

        System.out.println(temp+" is not a Palindrome
no.");
    }
}

```

OUTPUT:

```

Enter a No : 12521
12521 is a Palindrome no.

```

8) Write a java program to identify whether a number is a perfect number or not.

```

import java.util.Scanner;
public class Perfect
{
    public static void main(String [] args)
    {
        Scanner sc = new Scanner(System.in);
        int n,i,sum=0;
        System.out.print("Enter a No : ");
        n = sc.nextInt();
        for(i=1;i<n;i++)
        {
            if(n%i == 0)
                sum+=i;
        }
        if(sum == n)
            System.out.println(n+" is a Perfect no.");
        else

            System.out.println(n+" is not a Perfect no.");
    }
}

```



```
    }  
}
```

OUTPUT:

Enter a No : 6

6 is a Perfect no.

9) Write a java program to search for an element in an array using linear search.

```
import java.util.Scanner;  
public class LinearSearch  
{  
    public static void main(String [] args)  
    {  
        Scanner sc = new Scanner(System.in);  
        int size,i,find,ans=0;  
        System.out.print("Enter the Size of the Array: ");  
        size = sc.nextInt();  
        int [] a = new int[size];  
        for(i=0;i<size;i++)  
        {  
            System.out.print("Enter "+(i+1)+" Element: ");  
            a[i] = sc.nextInt();  
        }  
        System.out.print("Enter a Element to search in the  
Array: ");  
        find = sc.nextInt();  
        for(int x:a)  
        {  
            if(x == find)
```

```

        {
            ans++;
            break;
        }
    }
    if(ans==1)
        System.out.println(find+" is present in the
array");
    else

        System.out.println(find+" is not present in the
array");
    }
}

```

OUTPUT:

```

Enter the Size of the Array: 4
Enter 1 Element: 54
Enter 2 Element: 22
Enter 3 Element: 8
Enter 4 Element: -9
Enter a Element to search in the Array: 8
8 is present in the array

```

10) Write a java program to search for an element in an array using binary search.

```

import java.util.Scanner;
public class BinarySearch
{
    public static void main(String [] args)

```

```

{
    Scanner sc = new Scanner(System.in);
    int size,i,find,ans=0,start,end,middle;
    System.out.print("Enter the Size of the Array: ");
    size = sc.nextInt();
    int [] a = new int[size];
    for(i=0;i<size;i++)
    {
        System.out.print("Enter "+(i+1)+" Element: ");
        a[i] = sc.nextInt();
    }
    System.out.print("Enter a Element to search in the
Array: ");
    find = sc.nextInt();
    start = 0;
    end = size-1;
    while(start<=end)
    {
        middle = (start+end)/2;
        if(a[middle] == find)
        {
            System.out.println(find+" is present in the
array at "+(middle+1)+" position");
            ans++;
            break;
        }
        if(a[middle] > find)
            end = middle-1;
        if(a[middle] < find)
            start = middle+1;
    }
    if(ans==0)

```

```
        System.out.println(find+" is not present in the  
array");  
    }  
}
```

OUTPUT:

```
Enter the Size of the Array: 4  
Enter 1 Element: 1  
Enter 2 Element: 2  
Enter 3 Element: 3  
Enter 4 Element: 4  
Enter a Element to search in the Array: 4  
4 is present in the array at 4 position
```

11) Write a java program to sort an array of elements using bubble sort.

```
import java.util.Scanner;  
public class BubbleSort  
{  
    public static void main(String [] args)  
    {  
        Scanner sc = new Scanner(System.in);  
        int size,i,j,temp;  
        System.out.print("Enter the Size of the Array: ");  
        size = sc.nextInt();  
        int [] a = new int[size];  
        for(i=0;i<size;i++)  
        {  
            System.out.print("Enter "+(i+1)+" Element: ");  
            a[i] = sc.nextInt();  
        }  
    }  
}
```

```

    }
    System.out.println("Array before Sorting: ");
    for(int x:a)
        System.out.print(x+"\t");
    for(i=0;i<size-1;i++)
    {
        for(j=0;j<size-1-i;j++)
        {
            if(a[j]>a[j+1])
            {
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
        }
    }
    System.out.println("\nArray after Sorting: ");
    for(int x:a)
        System.out.print(x+"\t");

}
}

```

OUTPUT:

Enter the Size of the Array: 5

Enter 1 Element: -9

Enter 2 Element: 4

Enter 3 Element: 8

Enter 4 Element: 16

Enter 5 Element: 0

Array before Sorting:

-9 4 8 16 0

Array after Sorting:

-9 0 4 8 16

12) Write a java program for matrix addition.

```
import java.util.Scanner;
public class MatrixAddition
{
    public static void main(String [] args)
    {
        Scanner sc = new Scanner(System.in);
        int ar,ac,br,bc,i,j;

        //Matrix-1
        System.out.print("Enter the Rows and Columns of the
First Matrix: ");
        ar = sc.nextInt();
        ac = sc.nextInt();
        int [][] a = new int[ar][ac];
        for(i=0;i<ar;i++)
        {
            for(j=0;j<ac;j++)
            {
                System.out.print("Enter"+(i+1)+" "+(j+1)+"
Element: ");
                a[i][j] = sc.nextInt();
            }
        }

        //Matrix-2
```

```

        System.out.print("Enter the Rows and Columns of the
Second Matrix: ");
        br = sc.nextInt();
        bc = sc.nextInt();
        int [][] b = new int[br][bc];
        for(i=0;i<br;i++)
        {
            for(j=0;j<bc;j++)
            {
                System.out.print("Enter"+(i+1)+" , "+(j+1)+"
Element: ");

                b[i][j] = sc.nextInt();
            }
        }

//Printing of Matrix-1
System.out.println("First Matrix: ");
for(i=0;i<ar;i++)
{
    for(j=0;j<ac;j++)
    {
        System.out.print(a[i][j]+"\\t");
    }
    System.out.println();
}

//Printing of Matrix-2
System.out.println("Second Matrix: ");
for(i=0;i<br;i++)
{
    for(j=0;j<bc;j++)
    {

```

```

        System.out.print(b[i][j]+"\\t");
    }
    System.out.println();
}

//Addition of Matrix
if(ar==br && ac==bc)
{
    int [][] ans = new int [ar][bc];
    for(i=0;i<br;i++)
    {
        for(j=0;j<bc;j++)
        {
            ans[i][j] = a[i][j] + b[i][j];
        }
    }

    //Printing the sum of the matrices
    System.out.println("The sum of Two Matrices is:
");

    for(i=0;i<ar;i++)
    {
        for(j=0;j<bc;j++)
        {
            System.out.print(ans[i][j]+"\\t");
        }
        System.out.println();
    }
}
else
    System.out.println("Addition of the Matrices is
not possible");

```



```
    }  
}
```

OUTPUT:

Enter the Rows and Columns of the First Matrix: 2 3

Enter1,1 Element: 1

Enter1,2 Element: 2

Enter1,3 Element: 3

Enter2,1 Element: 4

Enter2,2 Element: 5

Enter2,3 Element: 6

Enter the Rows and Columns of the Second Matrix: 2 3

Enter1,1 Element: 5

Enter1,2 Element: 5

Enter1,3 Element: 5

Enter2,1 Element: 5

Enter2,2 Element: 5

Enter2,3 Element: 5

First Matrix:

1	2	3
---	---	---

4	5	6
---	---	---

First Matrix:

5	5	5
---	---	---

5	5	5
---	---	---

The sum of Two Matrices is:

6	7	8
---	---	---

9	10	11
---	----	----

13) Write a java program for matrix multiplication.

```
import java.util.Scanner;
public class MatrixMul
{
    public static void main(String [] args)
    {
        Scanner sc = new Scanner(System.in);
        int ar,ac,br,bc,i,j,k;

        //Matrix-1
        System.out.print("Enter the Rows and Columns of the
First Matrix: ");
        ar = sc.nextInt();
        ac = sc.nextInt();
        int [][] a = new int[ar][ac];
        for(i=0;i<ar;i++)
        {
            for(j=0;j<ac;j++)
            {
                System.out.print("Enter" + (i+1) + ", " + (j+1) + "
Element: ");
                a[i][j] = sc.nextInt();
            }
        }

        //Matrix-2
        System.out.print("Enter the Rows and Columns of the
Second Matrix: ");
        br = sc.nextInt();
        bc = sc.nextInt();
```

```

int [][] b = new int[br][bc];
for(i=0;i<br;i++)
{
    for(j=0;j<bc;j++)
    {
        System.out.print("Enter"+(i+1)+" "+(j+1)+"
Element: ");
        b[i][j] = sc.nextInt();
    }
}

//Printing of Matrix-1
System.out.println("First Matrix: ");
for(i=0;i<ar;i++)
{
    for(j=0;j<ac;j++)
    {
        System.out.print(a[i][j]+"\\t");
    }
    System.out.println();
}

//Printing of Matrix-2
System.out.println("Second Matrix: ");
for(i=0;i<br;i++)
{
    for(j=0;j<bc;j++)
    {
        System.out.print(b[i][j]+"\\t");
    }
    System.out.println();
}

```

```

//Multiplication of Matrix
if(ac == br)
{
    int [][] ans = new int [ar][bc];
    for(i=0;i<ar;i++)
    {
        for(j=0;j<bc;j++)
        {
            ans[i][j] = 0;
            for(k=0;k<ac;k++)
                ans[i][j] += a[i][k] * b[k][j];
        }
    }

    //Printing the Product of the matrices
    System.out.println("The Product of Two Matrices
is: ");

    for(i=0;i<ar;i++)
    {
        for(j=0;j<bc;j++)
        {
            System.out.print(ans[i][j]+"\\t");
        }
        System.out.println();
    }
}
else
    System.out.println("Multiplication of the
Matrices is not possible");
}
}

```

OUTPUT:

```
Enter the Rows and Columns of the First Matrix: 2 3
Enter1,1 Element: 1
Enter1,2 Element: 4
Enter1,3 Element: 5
Enter2,1 Element: 9
Enter2,2 Element: 0
Enter2,3 Element: 7
Enter the Rows and Columns of the Second Matrix: 3 1
Enter1,1 Element: 2
Enter2,1 Element: 8
Enter3,1 Element: 4
First Matrix:
1      4      5
9      0      7
Second Matrix:
2
8
4
The Product of Two Matrices is:
54
46
```

14) Write a java program to compute addition and multiplication of any two complex numbers.

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

```

{
    Scanner sc = new Scanner(System.in);
    int a_real, a_img, b_real, b_img;

    System.out.print("Enter Real and Imaginary part of
first complex no. : ");
    a_real = sc.nextInt();
    a_img = sc.nextInt();

    System.out.print("Enter Real and Imaginary part of
second complex no. : ");
    b_real = sc.nextInt();
    b_img = sc.nextInt();

    System.out.println("First complex nos. is: "+a_real+"
+ "+a_img+"i");

    System.out.println("Second complex nos. is: "+b_real+"
+ "+b_img+"i");

    //Addition
    int add_real , add_img;
    add_real = a_real + b_real;
    add_img = a_img + b_img;
    System.out.println("The Sum of the complex nos. is:
"+add_real+" + "+add_img+"i");

    //Multiplication
    int mul_real,mul_img;
    mul_real = (a_real * b_real) - (a_img * b_img);
    mul_img = (a_real * b_img) + (a_img * b_real);

```

```

        System.out.println("The Product of the complex nos.
is: "+mul_real+" + "+mul_img+"i");
    }
}

```

OUTPUT:

```

Enter Real and Imaginary part of first complex no. : 2 3
Enter Real and Imaginary part of second complex no. : 4 5
First complex nos. is: 2 + 3i
Second complex nos. is: 4 + 5i
The Sum of the complex nos. is: 6 + 7i
The Product of the complex nos. is: -7 + 22i

```

15) Write a java program to find the greatest common divisor of any two numbers.

```

import java.util.Scanner;
public class Gcd
{
    public static void main(String [] args)
    {
        Scanner sc = new Scanner(System.in);
        int a,b,start,i,ans=0;
        System.out.print("Enter 2 Numbers: ");
        a = sc.nextInt();
        b = sc.nextInt();
        if(a>b)
            start = b;
        else
            start = a;
        for(i=start;i>=1;i--)

```

```

        {
            if(a%i==0 && b%i==0)
            {
                ans = i;
                break;
            }
        }
        System.out.println("GCD of "+a+" and "+b+" is: "+ans);
    }
}

```

OUTPUT:

Enter 2 Numbers: 18 24

GCD of 18 and 24 is: 6

16) Write a java program to compute the sum of the digits of a given number.

```

import java.util.Scanner;
public class SumOfDigits
{
    public static void main(String [] args)
    {
        Scanner sc = new Scanner(System.in);
        int n,temp,sum=0;
        System.out.print("Enter a No : ");
        n = sc.nextInt();
        temp = n;
        while(n>0)
        {
            sum += n%10;

```



```

        n = n/10;
    }
    System.out.println("Sum of Digits of "+temp+" is:
"+sum);
    }
}

```

OUTPUT:

Enter a No : 9125

Sum of Digits of 9125 is: 17

17) Write a java program to display the default values of all primitive data types.

```

public class DefaultValues
{
    static byte b;
    static short s;
    static int i;
    static long l;
    static float f;
    static double d;
    static boolean bool;
    static char c;

    public static void main(String[] args)
    {
        System.out.println("Default values of primitive data
types:");
        System.out.println("byte: " + b);
    }
}

```

```

        System.out.println("short: " + s);
    System.out.println("int: " + i);
        System.out.println("long: " + l);
    System.out.println("float: " + f);
        System.out.println("double: " + d);
    System.out.println("boolean: " + bool);
        System.out.println("char: " + c);
    }
}

```

OUTPUT:

Default values of primitive data types:

byte: 0

short: 0

int: 0

long: 0

float: 0.0

double: 0.0

boolean: false

char:

18) Write a java program to find out the roots of a quadratic equation.

```

import java.util.Scanner;
import java.lang.Math;
public class QuadraticEquation
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
    }
}

```

```

        double a, b, c, D, root1, root2, root, realPart,
imaginaryPart;

        System.out.print("Enter the coefficient a: ");
        a = sc.nextDouble();

        System.out.print("Enter the coefficient b: ");
        b = sc.nextDouble();

        System.out.print("Enter the coefficient c: ");
        c = sc.nextDouble();

        D = (b * b) - (4 * a * c);

        if (D > 0)
        {
            root1 = (-b + Math.sqrt(D)) / (2 * a);
            root2 = (-b - Math.sqrt(D)) / (2 * a);
            System.out.println("The roots are " + root1
+ " and " + root2);
        }
        else if (D == 0)
        {
            root = -b / (2 * a);
            System.out.println("The root is " + root);
        }
        else
        {
            realPart = -b / (2 * a);
            imaginaryPart = Math.sqrt(-D) / (2 * a);

```

```

        System.out.println("The roots are " +
realPart + " + " + " + imaginaryPart + "i" + " and " + realPart + "
- " + imaginaryPart + "i");
    }
}
}

```

OUTPUT:

```

Enter the coefficient a: 1
Enter the coefficient b: 4
Enter the coefficient c: 8
The roots are -2.0 + 2.0i and -2.0 - 2.0i

```

19) Write a java program to find the sum of first 'n' natural numbers using a for-each loop.

```

import java.util.Scanner;
public class SumNaturalNos
{
    public static void main(String [] args)
    {
        Scanner sc = new Scanner(System.in);
        int n,i,sum=0;
        System.out.print("Enter the n Value: ");
        n = sc.nextInt();
        int [] temp = new int[n];
        for(i=1;i<=n;i++)
        {
            temp[i-1] = i;
        }
        for(int x:temp)

```

```

        {
            sum += x;
        }
        System.out.print("The sum of first "+n+" natural Nos.
is: "+sum);
    }
}

```

OUTPUT:

Enter the n Value: 8

The sum of first 8 natural Nos. is: 36

20) Write a java program to check whether the one dimensional array has duplicate elements or not.

```

import java.util.Scanner;
public class CheckDuplicates
{
    public static void main(String [] args)
    {
        Scanner sc = new Scanner(System.in);
        int size,i,j,count=0;
        System.out.print("Enter array size : ");
        size = sc.nextInt();
        int a[] = new int[size];
        for (i=0;i<size;i++)
        {
            System.out.print("Enter "+(i+1)+"th index
element : ");
            a[i] = sc.nextInt();
        }
    }
}

```

```

        for(i=0;i<size;i++)
        {
            for(j=i;j<size-1;j++)
            {
                if(a[i] == a[j+1])
                {
                    count++;
                    break;
                }
            }
        }
        if(count != 0)
            System.out.println("Has Duplicate Elements");
        else
            System.out.println("Does not have Duplicate
Elements");
    }
}

```

OUTPUT:

```

Enter array size : 5
Enter 1th index element : 1
Enter 2th index element : 2
Enter 3th index element : 1
Enter 4th index element : 4
Enter 5th index element : 4
Has Duplicate Elements

```

21) Write a java program to print duplicate elements of one dimensional array.

```
import java.util.Arrays;
import java.util.Scanner;
public class DuplicatesPrinting
{
    public static void main(String[] args)
    {
        int n;
        boolean hasDups = false;
        //Taking input
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size : ");
        n = sc.nextInt();
        int a[] = new int[n]; //creating array of length n
        for (int i = 0; i < n; i++)
        {
            System.out.print("Enter " + (i+1) + "th index element :
");
            a[i] = sc.nextInt();
        }
        //printing duplicates
        System.out.println("Duplicates are : ");
        Arrays.sort(a);
        int c = 1;
        for (int i = 1; i < a.length; i++)
        {
            if(a[i]==a[i-1])
            {
                hasDups = true;
            }
        }
    }
}
```

```

        c++;
        if(c==2)
        {
            System.out.print(a[i]+"\\t");
        }
    }
    else
    {
        c=1;
    }
}
//in case if there are no duplicates :
if (hasDups == false)
{
    System.out.println("No duplicate elements");
}
}
}

```

OUTPUT:

```

Enter array size : 6
Enter 1th index element : 1
Enter 2th index element : 4
Enter 3th index element : 1
Enter 4th index element : 1
Enter 5th index element : 2
Enter 6th index element : 2
Duplicates are :
1      2

```


22) Write a java program to insert an element into a specific position of the given array.

```
import java.util.Scanner;
public class InsertElement
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int n,i,index;
        System.out.print("Enter array size: ");
        n = sc.nextInt();
        int arr[] = new int[n+1];
        System.out.print("Enter "+n+" elements: ");
        for (i = 0; i <n; i++)
        {
            arr[i] = sc.nextInt();
        }
        System.out.println("Array Elements are: ");
        for(i = 0;i <n; i++)
        {
            System.out.print(arr[i]+"\\t");
        }
        System.out.println();
        System.out.print("Enter index to add element: ");
        index = sc.nextInt();
        i = n;
        while(i>index)
        {
            arr[i] = arr[i-1];
            i--;
        }
    }
}
```

```

    }
    System.out.print("Enter element to insert: ");
    arr[index] = sc.nextInt();
    System.out.println("Array after inserting: ");
    for (int x : arr)
    {
        System.out.print(x + "\t");
    }
    System.out.println();
}
}

```

OUTPUT:

Enter array size: 5

Enter 5 elements: 1 2 3 4 5

Array Elements are:

1 2 3 4 5

Enter index to add element: 2

Enter element to insert: -9

Array after inserting:

1 2 -9 3 4 5

23) Write a java program to delete an element into a specific position of the given array.

```

import java.util.Scanner;
public class DeleteElement
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
    }
}

```

```

        System.out.print("Enter array size: ");
        int n = sc.nextInt();
        int arr[] = new int[n];
        System.out.print("Enter "+n+" elements: ");
        for(int i = 0; i <n; i++)
        {
            arr[i] = sc.nextInt();
        }
        for(int x : arr)
        {
            System.out.print(x + "\t");
        }
        System.out.println();
        System.out.print("Enter index to remove element: ");
        int index = sc.nextInt();
        int i =index+1;
        while(i<n)
        {
            arr[i-1]=arr[i];
            i++;
        }
        System.out.println("Array after deleting: ");
        for (int j = 0; j < arr.length-1; j++)
        {
            System.out.print(arr[j]+" \t");
        }
        System.out.println();
    }
}

```

OUTPUT:

Enter array size: 5

```
Enter 5 elements: 1 2 3 4 5
1           2           3           4           5
Enter index to remove element: 2
Array after deleting:
1           2           4           5
```

24) Write a java program to display the students details using setter and getter methods of student class.

```
import java.util.Scanner;
class StudentDetails
{
    int rno;
    String name;
    double cgpa;
    public void set_rno(int x)
    {
        rno = x;
    }
    public void set_name(String y)
    {
        name = y;
    }
    public void set_cgpa(double z)
    {
        cgpa = z;
    }
    public int get_rno()
    {
        return rno;
    }
}
```

```

    public String get_name()
    {
        return name;
    }
    public double get_cgpa()
    {
        return cgpa;
    }
    public void display()
    {
        System.out.println("Student Details:\nRoll no:
"+rno+"\tName: "+name+"\tCGPA: "+cgpa);
    }
}
public class Student
{
    public static void main(String [] args)
    {
        Scanner sc = new Scanner(System.in);
        StudentDetails s = new StudentDetails();
        System.out.print("Enter student roll no: ");
        int rno = sc.nextInt();
        s.set_rno(rno);
        System.out.print("Enter student Name: ");
        String name = sc.next();
        s.set_name(name);
        System.out.print("Enter student CGPA: ");
        Double cgpa = sc.nextDouble();
        s.set_cgpa(cgpa);
        System.out.println("Student roll no: "+s.get_rno());
        System.out.println("Student Name: "+s.get_name());
        System.out.println("Student CGPA: "+s.get_cgpa());
    }
}

```

```
        s.display();
    }
}
```

OUTPUT:

```
Enter student roll no: 101
Enter student Name: Naina
Enter student CGPA: 9.5
Student roll no: 101
Student Name: Naina
Student CGPA: 9.5
Student Details:
Roll no: 101      Name: Naina      CGPA: 9.5
```

25) Write a java program to compute the volume of a cube using method return type and formal parameters.

```
import java.util.Scanner;
class Box
{
    double a;
    public Box()
    {
        a = 10.5;
    }
    public double volume(double s)
    {
        return s*s*s;
    }
}
public class BoxVolume
```

```

{
    public static void main(String[] args)
    {
        Box b = new Box();
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the edge of cube: ");
        double edge = sc.nextDouble();
        System.out.println("The volume of cube is: " +
b.volume(edge));
    }
}

```

OUTPUT:

Enter the edge of cube: 3.5

The volume of cube is: 42.875

26) Write a java program to demonstrate default constructor using box class and compute volume of it.

```

class Box
{
    double l,w,h;
    public Box()
    {
        l = 5.2;
        w = 2.4;
        h = 1.5;
    }
    public double volume()
    {
        return l*w*h;
    }
}

```

```

    }
}
public class DefaultConstructor
{
    public static void main(String [] args)
    {
        Box b = new Box();
        System.out.println("Volume is: "+b.volume());
    }
}

```

OUTPUT:

Volume is: 18.72

27) Write a java program to demonstrate parameterized constructor using box class and compute volume of it.(Note: Read data dynamically from the user)

```

import java.util.Scanner;
class Box
{
    double l,b,w;
    Box(double l,double b,double w)
    {
        this.l = l;
        this.b = b;
        this.w = w;
    }
    public double volume()
    {
        return l*b*w;
    }
}

```



```

    }
}
class ParameterizedConstructor
{
    public static void main(String [] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter l,b,w of a cuboid: ");
        double l = sc.nextDouble();
        double b = sc.nextDouble();
        double w = sc.nextDouble();
        Box obj = new Box(l,b,w);
        System.out.println("Volume is: "+obj.volume());
    }
}

```

OUTPUT:

Enter l,b,w of a cuboid: 2.5 6.3 4.1

Volume is: 64.57499999999999

28) Write a java program to demonstrate constructor overloading.

```

import java.util.Scanner;
class Box
{
    private double l,b,w;
    public Box()
    {
        l = b = w = 10.0;
    }
    Box(double l,double b,double w)

```

```

        {
            this.l = l;
            this.b = b;
            this.w = w;
        }
        public double volume()
        {
            return l*b*w;
        }
    }
}

class ConstructorOverloading
{
    public static void main(String [] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter l,b,w of a cuboid: ");
        double l = sc.nextDouble();
        double b = sc.nextDouble();
        double w = sc.nextDouble();
        Box b1 = new Box();
        System.out.println("Volume(Default Values) is:
"+b1.volume());
        Box b2 = new Box(l,b,w);
        System.out.println("Volume(User Values) is:
"+b2.volume());
    }
}

```

OUTPUT:

Enter l,b,w of a cuboid: 1.5 3.3 6.2

Volume(Default Values) is: 1000.0

Volume(User Values) is: 30.689999999999998

29) Write a java program to perform stack operations using arrays.

```
import java.util.Scanner;
class Stack
{
    int top=-1;
    int arr[];
    public Stack(int s)
    {
        arr = new int[s];
    }
    public boolean isFull()
    {
        if (top ==arr.length-1)
            return true;
        return false;
    }
    public boolean isEmpty()
    {
        if (top ==-1)
            return true;
        return false;
    }
    public void push(int ele)
    {
        if(!isFull())
        {
            top++;
            arr[top] = ele;
        }
    }
}
```

```

    }
    else
        System.out.println("stack is full !!! Can't
insert");
    }
    public int pop()
    {
        if(!isEmpty())
        {
            int e = arr[top];
            arr[top]=0;
            top--;
            return e;
        }
        else
        {
            System.out.println("stack is empty");
            return 0;
        }
    }
    public int peak()
    {
        if(!isEmpty())
        {
            return arr[top];
        }
        else
        {
            System.out.println("stack is empty");
            return 0;
        }
    }
}

```

```

    public void display(int size)
    {
        System.out.println("Stack elements are: ");
        for (int i = size; i>=0; i--)
            System.out.println(arr[i]+" ");
    }
}

public class StackUsingArray
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int size=-1;
        System.out.print("Enter stack size: ");
        int n =sc.nextInt();
        Stack s1 = new Stack(n);
        while(true)
        {
            System.out.println("Stack Operations
are:\n1) Push\n2) Pop\n3) Peak\n4) Display\n5) Exit");
            System.out.print("Choose an Operation: ");
            int choice = sc.nextInt();
            if(choice <1 || choice>5)
            {
                System.out.println("invalid input!");
            }
            if(choice == 1)
            {
                System.out.print("Enter element to
push: ");

                s1.push(sc.nextInt());
                size++;
            }
        }
    }
}

```

```

        }
        if(choice == 2)
        {
            System.out.println("Popped element:
"+s1.pop());

            size--;
        }
        if(choice == 3)
        {
            System.out.println("Peaked element
"+s1.peak());
        }
        if(choice == 4)
        {
            s1.display(size);
        }
        if(choice == 5)
        {
            System.out.println("Stack Closed");
            break;
        }
    }
}

```

OUTPUT:

Enter stack size: 5

Stack Operations are:

1) Push

2) Pop

3) Peak

4) Display

5)Exit

Choose an Operation: 1

Enter element to push: 10

Stack Operations are:

1)Push

2)Pop

3)Peak

4)Display

5)Exit

Choose an Operation: 1

Enter element to push: 20

Stack Operations are:

1)Push

2)Pop

3)Peak

4)Display

5)Exit

Choose an Operation: 1

Enter element to push: 30

Stack Operations are:

1)Push

2)Pop

3)Peak

4)Display

5)Exit

Choose an Operation: 2

Popped element: 30

Stack Operations are:

1)Push

2)Pop

3)Peak

4)Display

5)Exit

Choose an Operation: 3

Peaked element 20

Stack Operations are:

1)Push

2)Pop

3)Peak

4)Display

5)Exit

Choose an Operation: 4

Stack elements are:

20

10

Stack Operations are:

1)Push

2)Pop

3)Peak

4)Display

5)Exit

Choose an Operation: 5

Stack Closed

30) Write a java program to demonstrate copy constructor.

```
import java.util.Scanner;
class Box
{
    double length, breadth, height;
    public Box(double l, double b, double h)
    {
        length = l;
```



```

        breadth = b;
        height = h;
    }
    public Box(Box obj)
    {
        length = obj.length;
        breadth = obj.breadth;
        height = obj.height;
    }
    double volume()
    {
        return length * breadth * height;
    }
}

public class CopyConstructor
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter length,breadth,height of box:
");

        Box b1 = new
Box(sc.nextDouble(),sc.nextDouble(),sc.nextDouble());
        Box b2 = new Box(b1);
        System.out.println("Box 1 vol :"+b1.volume());
        System.out.println("Box 2 vol :"+b2.volume());
        b2.length =1;
        b2.breadth=5;
        b2.height=2;
        System.out.println("modified volume of box2
:"+b2.volume());
    }
}

```

```
}
```

OUTPUT:

Enter length,breadth,height of box: 1 2 3

Box 1 vol :6.0

Box 2 vol :6.0

modified volume of box2 :10.0

31) Write a java program to demonstrate Single Inheritance using box and cube class.

```
class Box
{
    double length, breadth, height;
    public Box()
    {
        length = -1;
        breadth = -1;
        height = -1;
    }
    public Box(double l, double b, double h)
    {
        length = l;
        breadth = b;
        height = h;
    }
    public void getDimensions()
    {
        System.out.println("length :"+length+"\nbreadth :"+breadth+"\nheight :"+height);
    }
}
```

```

    public String toString()
    {
        return "toString method of Box class";
    }
    double volume()
    {
        return length * breadth * height;
    }
}
class Cuboid extends Box
{
    Cuboid(double l , double b, double h)
    {
        super(l,b,h);
    }
    double volume()
    {
        return length * breadth * height;
    }
    public String toString()
    {
        System.out.println(super.toString());
        return "cuboid dimensions are "+length" "+breadth"
"+height;
    }
}
public class SingleInheritance
{
    public static void main(String[] args)
    {
        Box b = new Box();
        System.out.println("Dimensions of box are: ");
    }
}

```

```

        b.getDimensions();

        Cuboid c = new Cuboid(1,2,3);
        System.out.println("Dimensions of cuboid: ");
        c.getDimensions();

        System.out.println("volume of cuboid :"+c.volume());
        System.out.println(b+"\n"+c);
    }
}

```

OUTPUT:

Dimensions of box are:

length :-1.0

breadth :-1.0

height :-1.0

Dimensions of cuboid:

length :1.0

breadth :2.0

height :3.0

volume of cuboid :6.0

toString method of Box class

toString method of Box class

cuboid dimensions are 1.0 2.0 3.0

32) Write a java program to demonstrate Multilevel Inheritance using box, cube and cuboid class.

```
class Box
{
    protected double l,w,h;
    Box()
    {
        l = w = h = -1;
    }
    Box(double l,double w,double h)
    {
        this.l = l;
        this.w = w;
        this.h = h;
    }
    public void getDimensions()
    {
        System.out.println("Length: "+l+"\nWidth:
"+w+"\nHeight: "+h);
    }
}
class Cube extends Box
{
    Cube()
    {
        super();
    }
    Cube(double l)
    {
        this.l = this.w = this.h = l;
    }
}
```

```

    }
    public double volume()
    {
        return this.l * this.w * this.h;
    }
}
class Cuboid extends Cube
{
    Cuboid()
    {
        super();
    }
    Cuboid(double l,double w,double h)
    {
        this.l = l;
        this.w = w;
        this.h = h;
    }
    public double volume()
    {
        return this.l * this.w * this.h;
    }
}
public class MultiLevelInheritance
{
    public static void main(String[] args)
    {
        Box b = new Box();
        System.out.println("Dimensions of box are: ");
        b.getDimensions();

        Cube c = new Cube();
    }
}

```

```

        System.out.println("Dimensions of cube are: ");
        c.getDimensions();
        System.out.println("The volume of cube is:
"+c.volume());

        Cube c1 = new Cube(5);
        System.out.println("Dimensions of cube are: ");
        c1.getDimensions();
        System.out.println("The volume of cube is:
"+c1.volume());

        Cuboid d = new Cuboid();
        System.out.println("Dimensions of cuboid are: ");
        d.getDimensions();
        System.out.println("The volume of cuboid is:
"+d.volume());

        Cuboid d1 = new Cuboid(2,3,4);
        System.out.println("Dimensions of cuboid are: ");
        d1.getDimensions();
        System.out.println("The volume of cuboid is:
"+d1.volume());
    }
}

```

OUTPUT:

```

Dimensions of box are:
Length: -1.0
Width: -1.0
Height: -1.0
Dimensions of cube are:
Length: -1.0

```

```
Width: -1.0
Height: -1.0
The volume of cube is: -1.0
Dimensions of cube are:
Length: 5.0
Width: 5.0
Height: 5.0
The volume of cube is: 125.0
Dimensions of cuboid are:
Length: -1.0
Width: -1.0
Height: -1.0
The volume of cuboid is: -1.0
Dimensions of cuboid are:
Length: 2.0
Width: 3.0
Height: 4.0
The volume of cuboid is: 24.0
```

33) Write a java program to demonstrate Hierarchical Inheritance using different shape classes.

```
import java.lang.Math;
class Shape
{
    double l,b,r;
    Shape()
    {
        l = b = r = -1;
    }
}
```



```

        public void area()
        {
            System.out.print("The area of ");
        }
    }

class Square extends Shape
{
    Square()
    {
        super();
    }
    Square(double s)
    {
        this.l = this.b = s;
    }
    public void area()
    {
        super.area();
        System.out.println("Square is: "+this.l*this.b);
    }
}

class Rectangle extends Shape
{
    Rectangle()
    {
        super();
    }
    Rectangle(double l,double b)
    {
        this.l = l;
        this.b = b;
    }
}

```

```

        public void area()
        {
            super.area();
            System.out.println("Rectangle is: "+this.l*this.b);
        }
    }
class Circle extends Shape
{
    Circle()
    {
        super();
    }
    Circle(double r)
    {
        this.r = r;
    }
    public void area()
    {
        super.area();
        System.out.println("Circle is:
"+Math.PI*this.r*this.r);
    }
}
public class HierarchicalInheritance
{
    public static void main(String [] args)
    {
        Square s1 = new Square();
        s1.area();

        Square s2 = new Square(5);
        s2.area();
    }
}

```

```

        Rectangle r1 = new Rectangle();
        r1.area();

        Rectangle r2 = new Rectangle(3,4);
        r2.area();

        Circle c1 = new Circle();
        c1.area();

        Circle c2 = new Circle(7);
        c2.area();
    }
}

```

OUTPUT:

```

The area of Square is: 1.0
The area of Square is: 25.0
The area of Rectangle is: 1.0
The area of Rectangle is: 12.0
The area of Circle is: 3.141592653589793
The area of Circle is: 153.93804002589985

```

34) Write a java program to demonstrate Method overloading.

```

class Demo
{
    void test()
    {
        System.out.println("No Parameters");
    }
}

```

```

void test(int a)
{
    System.out.println("a: "+a);
}
void test(int a,int b)
{
    System.out.println("a and b are: "+a+" "+b);
}
double test(double a)
{
    System.out.println("Double a: "+a);
    return a*a;
}
}
class MethodOverloading
{
    public static void main(String [] args)
    {
        Demo obj = new Demo();
        double ans;
        obj.test();
        obj.test(10);
        obj.test(10,20);
        ans = obj.test(1.5);
        System.out.println("Result of obj test(1.5): "+ans);
    }
}

```

OUTPUT:

No Parameters

a: 10

a and b are: 10 20

Double a: 1.5

Result of obj test(1.5): 2.25

35) Write a java program to demonstrate Method overriding.

```
class Parent
{
    int i,j;
    Parent(int a,int b)
    {
        i=a;
        j=b;
    }
    void show()
    {
        System.out.println("i and j are: "+i+" "+j);
    }
}
class Child extends Parent
{
    int k;
    Child(int x,int y,int z)
    {
        super(x,y);
        k = z;
    }
    void show()
    {
        super.show();
        System.out.println("k: "+k);
    }
}
```

```

}
class MethodOverriding
{
    public static void main(String [] args)
    {
        Child obj = new Child(2,3,7);
        obj.show();
    }
}

```

OUTPUT:

i and j are: 2 3

k: 7

36) Write a java program to demonstrate Dynamic method dispatch.

```

class parent
{
    public void disp()
    {
        System.out.println("parent Disp()");
    }
    public void show()
    {
        System.out.println("parent Show()");
    }
}
class child extends parent
{
    public void show()

```

```

        {
            System.out.println("child Show()");
        }
    }
}

public class DynamicMethodDispatch
{
    public static void main(String [] args)
    {
        parent a = new parent();
        a.disp();
        a.show();
        child b = new child();
        b.disp();
        b.show();
        parent obj;
        obj = new child();
        obj.disp();
        obj.show();
    }
}

```

OUTPUT:

```

parent Disp()
parent Show()
parent Disp()
child Show()
parent Disp()
child Show()

```

37) Write a java program to demonstrate Abstract class.

```
abstract class Myabstract
{
    public void show()
    {
        System.out.println("Concrete show() method");
    }
    public abstract void display();
    public abstract void message();
}
class Implementation extends Myabstract
{
    public void display()
    {
        System.out.println("display() method implementation");
    }
    public void message()
    {
        System.out.println("Implementation class");
    }
}
public class AbstractDemo
{
    public static void main(String [] args)
    {
        Implementation m = new Implementation();
        m.show();
        m.display();
        m.message();
        Myabstract n = new Implementation();
    }
}
```



```

        System.out.println("Using reference variable");
        n.show();
        n.display();
        n.message();
    }
}

```

OUTPUT:

```

Concrete show() method
display() method implementation
Implementation class
Using reference variable
Concrete show() method
display() method implementation
Implementation class

```

38) Write a java program to demonstrate final at variable level.

```

class Sample
{
    public static final int COUNT = 10;
    public int getCount()
    {
        return COUNT;
    }
}

public class FinalVariableLevel
{
    public static void main(String [] args)
    {
        final int COUNT;
    }
}

```

```

        Sample s = new Sample();
        System.out.println("Count: "+s.getCount());
        System.out.println("Count from Sample Class:
"+Sample.COUNT);
        COUNT = 1000;
        System.out.println("Local Final Count: "+COUNT);
    }
}

```

OUTPUT:

Count: 10

Count from Sample Class: 10

Local Final Count: 1000

39) Write a java program to demonstrate final at method level.

```

class Final
{
    public final void show()
    {
        System.out.println("Final show()");
    }
}
class Sub extends Final
{
    void display()
    {
        System.out.println("This is sub class");
    }
}
public class FinalMethodLevel

```

```

{
    public static void main(String [] args)
    {
        Final f = new Final();
        System.out.println("Super class method");
        f.show();
        Sub s = new Sub();
        System.out.println("Super class method invocation
through base class");
        s.show();
        s.display();
    }
}

```

OUTPUT:

Super class method

Final show()

Super class method invocation through base class

Final show()

This is sub class

40) Write a java program to demonstrate final at class level.

```

final class parent
{
    private String name = "Sushma";
    public String getName()
    {
        return name;
    }
}

```

```

public class FinalClassLevel
{
    public static void main(String [] args)
    {
        parent p = new parent();
        System.out.println("Name: "+p.getName());
    }
}

```

OUTPUT:

Name: Sushma

41) Write a java program to demonstrate Interface.

```

interface Intral
{
    int MAX = 99;
    void show();
    public abstract String message();
}
class Test implements Intral
{
    @Override
    public String message()
    {
        return "hello world";
    }
    @Override
    public void show()
    {
        System.out.println("MAX :"+MAX);
    }
}

```

```

        System.out.println("show method implemented");
    }
}

public class InterfaceDemo
{
    public static void main(String[] args)
    {
        Test t = new Test();
        t.show();
        System.out.println("the message : "+t.message());
    }
}

```

OUTPUT:

MAX :99

show method implemented

the message : hello world

42) Write a java program to implement multiple inheritance using interfaces.

```

interface Intra1
{
    int m=90;
    void display();
}

interface Intra2
{
    int max=90;
    void show();
}

```

```

class Test implements Intra1,Intra2
{
    public void display()
    {
        System.out.println("display method");
    }
    public void show()
    {
        System.out.println("show method");
    }
}

public class MultipleInheritance_Interface
{
    public static void main(String[] args)
    {
        Test t = new Test();
        t.display();
        t.show();
        Intra1 m1;
        m1 = new Test();
        m1.display();
        Intra2 m2;
        m2 = new Test();
        m2.show();
    }
}

```

OUTPUT:

```

display method
show method
display method
show method

```

43) Write a java program to demonstrate Functional Interface.

```
@FunctionalInterface
interface Intral
{
    void show();
}

public class FuntionalInterfaceDemo implements Intral
{
    @Override
    public void show()
    {
        System.out.println("show method implemented");
    }
    void display()
    {
        System.out.println("display method");
    }
    public static void main(String[] args)
    {
        FuntionalInterfaceDemo obj = new
FuntionalInterfaceDemo();
        obj.show();
        obj.display();
    }
}
```

OUTPUT:

```
show method implemented
display method
```

44) Write a java program to demonstrate default and static methods in an Interface.

```
interface Intral
{
    int max=90;
    void show();
    default void greet()
    {
        System.out.println("Welcome to College");
    }
    public static String message()
    {
        return "You chose good college";
    }
}

public class DefaultStaticInInterface implements Intral
{
    public void show()
    {
        System.out.println("max is :"+max);
    }
    public static void main(String[] args)
    {
        DefaultStaticInInterface obj = new
DefaultStaticInInterface();
        obj.show();
        obj.greet();
        System.out.println("The message is :-
"+Intral.message());
    }
}
```



```
}
```

OUTPUT:

max is :90

Welcome to College

The message is :- You chose good college

45) Write a java program to demonstrate nested or inner class.

```
class OuterClass
{
    private int pin = 2332;
    class InnerClass
    {
        public void show()
        {
            System.out.println("the pin is :"+pin);
        }
    }
}

public class InnerClassDemo
{
    public static void main(String[] args)
    {
        OuterClass outObj = new OuterClass();
        OuterClass.InnerClass inObj = outObj.new InnerClass();
        inObj.show();
    }
}
```

OUTPUT:

the pin is :2332

46) Write a java program to demonstrate Method level inner class.

```
class MyOuter
{
    private int code = 100;
    void methodLocal()
    {
        class MyInner
        {
            public void show()
            {
                System.out.println("The code is
: "+code);
            }
        }
        MyInner obj = new MyInner();
        obj.show();
    }
}

public class MethodLevelInnerclass
{
    public static void main(String[] args)
    {
        MyOuter m = new MyOuter();
        m.methodLocal();
    }
}
```

OUTPUT:

The code is :100

47) Write a java program to demonstrate anonymous inner class with respect to concrete class.

```
class MyClass1
{
    MyClass1()
    {
        System.out.println("MyClass1 Constructor");
    }
    public void disp(){}
    public void methodlocal()
    {
        System.out.println("MyClass1 Method");
    }
}
class MyClass2
{
    MyClass2()
    {
        System.out.println("MyClass2 Constructor");
    }
    MyClass1 m1 = new MyClass1()
    {
        public void methodlocal()
        {
            System.out.println("This is anonymous
methodlocal");
        }
        public void disp()
```

```

        {
            System.out.println("This is anonymous disp");
        }
    };
    void method()
    {
        m1.methodlocal();
        m1.disp();
    }
}
public class AnonymousInnerClass
{
    public static void main(String [] args)
    {
        MyClass2 obj = new MyClass2();
        obj.method();
    }
}

```

OUTPUT:

MyClass1 Constructor

MyClass2 Constructor

This is anonymous methodlocal

This is anonymous disp

48) Write a java program to demonstrate anonymous inner classes with interface implementation.

```

@FunctionalInterface
interface Age
{

```

```

        int AGE = 21;
        void getAge();
    }
class Anonymous
{
    public static void main(String [] args)
    {
        Age obj = new Age()
        {
            public void getAge()
            {
                System.out.println("Age is: "+AGE);
            }
        };
        obj.getAge();
    }
}

```

OUTPUT:

Age is: 21

49) Write a java program to demonstrate Static nested classes.

```

public class StaticNestedClass
{
    static class Nested
    {
        public void disp()
        {
            System.out.println("i am from static nested
class");
        }
    }
}

```

```

        }
    }
    public static void main(String[] args)
    {
        StaticNestedClass.Nested obj = new
StaticNestedClass.Nested();
        obj.disp();
    }
}

```

OUTPUT:

i am from static nested class

50) Write a java program to check whether the given array is a mountain array or not.

```

import java.util.Scanner;
public class MountainArray
{
    public static boolean MountainArray(int arr[])
    {
        if (arr.length <= 2)
            return false;
        boolean fall = false;
        boolean raise = false;
        for (int i = 1; i < arr.length; i++)
        {
            if (arr[i - 1] < arr[i])
            {
                raise = true;
                if (fall == true)
                    return false;
            }
        }
    }
}

```

```

        }
        else if (arr[i - 1] > arr[i])
        {
            fall = true;
            if (raise == false)
                return false;
        }
        else
            return false;
    }
    if (raise && fall)
        return true;
    return false;
}

public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    int size,i;
    System.out.print("Enter array size: ");
    size = sc.nextInt();
    int a[] = new int[size];
    for(i=0;i<size;i++)
    {
        System.out.print("Enter "+(i+1)+" Element: ");
        a[i] = sc.nextInt();
    }
    System.out.println("Array elements are :");
    for(i=0;i<size;i++)
        System.out.print(a[i]+"\\t");
    boolean res = MountainArray(a);
    if (res)

```

```
        System.out.println("\nGiven array is a Mountain  
Array");  
    else  
        System.out.println("\nGiven array is not Mountain  
Array");  
    }  
}
```

OUTPUT:

Enter array size: 4

Enter 1 Element: 0

Enter 2 Element: 3

Enter 3 Element: 2

Enter 4 Element: 1

Array elements are :

0 3 2 1

Given array is a Mountain Array