



**JB INSTITUTE OF ENGG & TECHNOLOGY (AUTONOMOUS)**

## Lab Manual

On

## JAVA PROGRAMMING LANGUAGE

MCA II year I semester

By

**HARIKA Y**

Assistant Professor

MCA Department

# JAVA PROGRAMMING LANGUAGE

---

|    |                                                       |    |
|----|-------------------------------------------------------|----|
| 1  | Write a program for check given number even or odd    | 4  |
| 2  | Write a program for check given number prime or not   | 4  |
| 3  | Display n prime numbers                               | 5  |
| 4  | Check given number is Armstrong or not                | 6  |
| 5  | Check given num is perfect or not                     | 7  |
| 6  | Print Fibonacci series                                | 8  |
| 7  | Display twin prime number below given number          | 8  |
| 8  | Generate current bill based on units                  | 10 |
| 9  | Print Fibonacci numbers up to given number            | 11 |
| 10 | Sum of integers using StringTokenizer                 | 12 |
| 11 | Find smallest n largest number from given n list      | 13 |
| 12 | Calculate this series value $1/1*1+1/2*2+.....1/n*n$  | 14 |
| 13 | Calculate ncr function                                | 15 |
| 14 | Check given string palindrome or not                  | 16 |
| 15 | Create substring from main string                     | 16 |
| 16 | Find word frequency in the given string               | 17 |
| 17 | Find letter frequency in the given string             | 18 |
| 18 | Program for Matrix functions                          | 19 |
| 19 | Program for tower of Hanoi problem                    | 21 |
| 20 | Simple client program                                 | 23 |
| 21 | simple server program                                 | 25 |
| 22 | Count no. Of lines ,words ,characters from given file | 29 |
| 23 | Change specific character in the given file           | 30 |
| 24 | program for linear search                             | 31 |

## JAVA PROGRAMMING LANGUAGE

---

|    |                                                                       |    |
|----|-----------------------------------------------------------------------|----|
| 25 | <b>program for binary search</b>                                      | 32 |
| 26 | <b>program for bubble sort</b>                                        | 35 |
| 27 | <b>Program for insertion sort</b>                                     | 36 |
| 28 | <b>Program for selection sort</b>                                     | 38 |
| 29 | <b>Program for Quick sort</b>                                         | 40 |
| 30 | <b>Implementation of stack ADT</b>                                    | 42 |
| 31 | <b>Implementation of queue ADT</b>                                    | 46 |
| 32 | <b>Example of Thread implementation of program</b>                    | 49 |
| 33 | <b>Example of Mouse event program</b>                                 | 52 |
| 34 | <b>Write a program for textbox color change based on mouse events</b> | 55 |
| 35 | <b>Program for Key events</b>                                         | 57 |
| 36 | <b>Program for keyboard Listener</b>                                  | 59 |
| 37 | <b>Program for Creating a JTABLE</b>                                  | 61 |
| 38 | <b>Program for Creating JButton</b>                                   | 62 |
| 39 | <b>Program for Creating rectangle, square etc..</b>                   | 63 |
| 40 | <b>Program for implement a simple calculator</b>                      | 64 |
| 41 | <b>Create simple form using all AWT components.</b>                   | 70 |
| 42 | <b>Program for all swing components creation.</b>                     | 73 |
| 43 | <b>Program for Creating scroll bar</b>                                | 76 |
| 44 | <b>Program for Creating menu bar</b>                                  | 77 |
| 45 | <b>Program for Creating Choice Box</b>                                | 78 |
| 46 | <b>Program for Creating List Box</b>                                  | 79 |
| 47 | <b>Program for Creating Check Box</b>                                 | 80 |

# JAVA PROGRAMMING LANGUAGE

---

## 1. Write a program for check given number even or odd

```
class Even
{
    public static void main(String arg[])
    {
        int a;
        a=Integer.parseInt(arg[0]);
        if(a%2==0)
            System.out.println("Given no is Even");
        else
            System.out.println("Given no is Odd");
    }
}
```

## 2. Write a program for check given number prime or not

```
class Prime
{
    public static void main(String arg[])
    {
        int a,count=1;
        a=Integer.parseInt(arg[0]);
        for(int i=0;i<a;i++)
        {
```

```
if(a%2==0)

count++;

}

if(count==2)

System.out.println("Given no is Prime");

else

System.out.println("Given no is not Prime");

}

}
```

### 3. Display n prime numbers

```
class Nprime

{

public static void main(String arg[])

{

int n,count=1;

n=Integer.parseInt(arg[0]);

for(int i=2;i<=n;i++)

{

count=1;

for(int j=2;j<=i;j++)

{

if(i%j==0)

count++;

}

}
```

```
if(count==2)
    System.out.println(i);}}
```

#### 4. Check given number is Armstrong or not

```
class Armstrong
{
    public static void main(String arg[])
    {
        int n,num,r,sum=0;
        num=Integer.parseInt(arg[0]);
        n=num;
        while(num!=0)
        {
            r=num%10;
            sum=sum+(r*r*r);
            num=num/10;
        }
        if(sum==n)
            System.out.println("The given no is Armstrong");
        else
            System.out.println("The given no is not Armstrong");
    }
}
```

## 5. Check given num is perfect or not

```
class Perfect
{
    public static void main(String arg[])
    {
        int n,i,res=0;
        n=Integer.parseInt(arg[0]);
        for(i=1;i<n;i++)
        {
            if(n%i==0)
                res=res+i;
        }
        if(res==n)
            System.out.println("The given no is Perfect");
        else
            System.out.println("The given no is not Perfect");
    }
}
```

## 6. Print Fibonacci series

```
class Fibonacci
{
    public static void main(String arg[])
    {
        int a=0,b=1,c=0,d;
        d=Integer.parseInt(arg[0]);
        System.out.println(a+b);
        c=a+b;
        while(c<d)
        {
            System.out.println(c);
            a=b;
            b=c;
            c=a+b;
        }
    }
}
```

## 7. Display twin prime number below given number

```
class Tprime
{
    public static void main(String arg[])
```

# JAVA PROGRAMMING LANGUAGE

---

```
{  
int n1,n2,count1=1,count2=1;  
n1=Integer.parseInt(arg[0]);  
n2=Integer.parseInt(arg[1]);  
if(n1-n2==2 || n1-n2== -2)  
{  
for(int i=2;i<=n1;i++)  
{  
if(n1%i==0)  
count1++;  
}  
for(int j=2;j<=n2;j++)  
{  
if(n2%j==0)  
count2++;  
}  
}  
if(count1==2 && count2==2)  
System.out.println("Given no is Twin Primes");  
else  
System.out.println("Given no is not Twin Primes");  
}  
}
```

## 8. Generate current bill based on units

```
class Bill
{
    public static void main(String arg[])
    {
        int n;
        double bill;
        n=Integer.parseInt(arg[0]);
        if(n<=100)
        {
            bill=n*1.25;
            System.out.println(bill);
        }
        else if(n>=100 && n<=200)
        {
            bill=n*2.5;
            System.out.println(bill);
        }
        else
        {
            bill=n*3.75;
```

```
System.out.println(bill);  
    }  
}
```

## 9. Print Fibonacci numbers up to given number

```
class Nfibonacci  
{  
    public static void main(String arg[])  
    {  
        int a=0,b=1,c=0,d;  
        d=Integer.parseInt(arg[0]);  
        System.out.println(a);  
        System.out.println(b);  
        c=a+b;  
        for(int i=0;i<(d-2);i++)  
        {  
            System.out.println(c);  
            a=b;  
            b=c;  
            c=a+b;  
        }  
    }  
}
```

## 10. Sum of integers using StringTokenizer

```
import java.util.*;

public class SumOfTokens

{

    public static void main(String arg[])

    {

        int n,sum=0,i=0;

        n=arg.length;

        int []a=new int[n];

        StringTokenizer st=new StringTokenizer("1 2 3 4 5");

        while(st.hasMoreElements())

        {

            a[i]=Integer.parseInt(st.nextElement());

            System.out.println(a[i]);

            sum=sum+a[i];

            i++;

        }

        System.out.println(sum);

    }

}
```

## 11. Find smallest n largest number from given list

```
class SmallLarge
{
    public static void main(String arg[])
    {
        int l,s,n;
        n=Integer.parseInt(arg[0]);
        int a[]=new int[n];
        for(int i=0,j=1;i<5;i++,j++)
        {
            a[i]=Integer.parseInt(arg[j]);
        }
        l=s=a[0];
        for(int i=0;i<5;i++)
        {
            if(a[i]>l)
            {
                l=a[i];
            }
            if(a[i]<s)
            {
                s=a[i];
            }
        }
    }
}
```

```
}  
  
}  
  
System.out.println("large no is"+l);  
  
System.out.println("small no is"+s);  
  
}  
  
}
```

**12. Calculate this series value  $1/1*1+1/2*2+.....1/n*n$**

```
class Series1  
{  
    public static void main(String arg[])  
    {  
        int n;  
        double sum=0;  
        n=Integer.parseInt(arg[0]);  
        for(int i=1;i<=n;i++)  
        {  
            sum=sum+(1/i*i);  
        }  
        System.out.println("Sum of series"+sum);  
    }  
}
```

## 13. Calculate ncr function

```
class Ncr
{
    public static void main(String arg[])
    {
        int n=5,r=3,ncr;
        ncr=fact(n)/((fact(r))*fact(n-r));
        System.out.println("n+c"+r"+"= "+ncr);
    }
}

public static int fact(int n)
{
    int i,t=1;
    for(i=0;i<=n;i++)
    {
        t=t*i;
    }
    return t;
}
```

## 14. Check given string palindrome or not

```
class StringPalindrome
```

```
{
```

```
public static void main(String args[])
{
    String str,temp;
    StringBuffer sb1=new StringBuffer("madam");
    str=sb1.toString();
    sb1.reverse();
    temp=sb1.toString();
    if(str.equals(temp))
        System.out.println("Given string is palindrome");
    else
        System.out.println("Given string isnot palindrome");
}
}
```

## **15. Create substring from main string**

```
import java.util.*;

class Sub
{
    public static void main(String arg[])
    {
        StringBuffer s1=new StringBuffer ("dis cherry");
        StringBuffer s2=new StringBuffer ("is");
        int n=s1.length()+s2.length();
        char[] c=new char[n];
        s1.getChars(0,3,c,0);
```

```
s2.getChars(0,2,c,4);  
s1.getChars(4,10,c,6);  
for(int i=0;i<n;i++)  
System.out.print(c[i]);  
}  
}
```

### 16. Find word frequency in the given string

```
import java.util.*;  
  
class WordFrequency  
{  
    public static void main(String arg[])  
    {  
        StringTokenizer s=new StringTokenizer("dis is cherry is sharath");  
  
        String s1="is";  
  
        int c=0;  
  
        while(s.hasMoreTokens())  
        {  
            String s2=s.nextToken();  
  
            if(s1.equals(s2))  
  
                c++;  
        }  
  
        System.out.println("Word Frequency is"+c);  
    }  
}
```

## 17. Find letter frequency in the given string

```
import java.util.*;

class LetterFrequency
{
    public static void main(String arg[])
    {
        String s=new String("dis is cherry is sharath");
        char c='s';
        char a[];
        a=s.toCharArray();
        int count=0;
        for(int i=0;i<a.length;i++)
        {
            if(a[i]==c)
            count++;
        }
        System.out.println("Letter Frequency is "+count);
    }
}
```

## 18. Program for Matrix functions

```
import java.util.*;

import java.io.*;

public class MatrixEx

{

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

    {

        String n;

        int m1[][]=new int[2][2];

        int m2[][]=new int[2][2];

        int m3[][]=new int[2][2];

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Enter matrix m1 elements");

        for(int i=0;i<2;i++)

        {

            for(int j=0;j<2;j++)

            {

                n=br.readLine();

                m1[i][j]=Integer.parseInt(n);

            }

        }

        System.out.println("Enter matrix m2 elements");
```

```
for(int i=0;i<2;i++)
{
for(int j=0;j<2;j++)
{
n=br.readLine();
m2[i][j]=Integer.parseInt(n);
}
}

System.out.println("matrix addition is");

for(int i=0;i<2;i++)
{
for(int j=0;j<2;j++)
{
m3[i][j]=m1[i][j]+m2[i][j];
}
}

for(int i=0;i<2;i++)
{
for(int j=0;j<2;j++)
{
System.out.print(m3[i][j]+" ");
}

System.out.println();
}
}
```

```
System.out.println("matrix Substraction is");
```

```
for(int i=0;i<2;i++)
```

```
{
```

```
for(int j=0;j<2;j++)
```

```
{
```

```
m3[i][j]=m1[i][j]-m2[i][j];
```

```
}
```

```
}
```

```
for(int i=0;i<2;i++)
```

```
{
```

```
for(int j=0;j<2;j++)
```

```
{
```

```
System.out.print(m3[i][j]+" ");
```

```
}
```

```
System.out.println();
```

```
}
```

```
}
```

```
}
```

### **19. Program for tower of Hanoi problem**

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

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

```
class Tower
```

```
{
```

```
public static void main(String arg[]) throws IOException
```

## JAVA PROGRAMMING LANGUAGE

---

```
{  
    int n;  
    char t1,t2,t3;  
    t1='s';  
    t2='m';  
    t3='l';  
    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));  
    System.out.println("Enter how many disks you want");  
    n=Integer.parseInt(br.readLine());  
    towerhonio(n,t1,t2,t3);  
}  
static void towerhonio(int n,char t1,char t2,char t3)  
{  
    if(n>0)  
    {  
        towerhonio(n-1,t1,t3,t2);  
        System.out.println(n+"Disk moves from"+"t1"+" "+"t3");  
        towerhonio(n-1,t3,t2,t1);  
    }  
}  
}
```

## 20. Simple client program

```
import java.io.*;

import java.net.*;

public class SimpleClient {

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

        Socket clientSocket = null;

        try {

            clientSocket = new Socket(args[0], 4442);

        } catch (UnknownHostException e) {

            System.err.println("Don't know about host: " + args[0] + ".");

            System.exit(1);

        } catch (IOException e) {

            System.err.println("Couldn't get I/O for " +

                "the connection to: " + args[0] + "");

            System.exit(1);

        }

    }

}
```

```
    BufferedInputStream in;
```

```
BufferedOutputStream out;
```

```
try {
```

```
    in = new BufferedInputStream(dientSocket.getInputStream());
```

```
    out = new BufferedOutputStream(dientSocket.getOutputStream());
```

```
} catch (IOException e) {
```

```
    System.out.println(e.toString());
```

```
    return;
```

```
}
```

```
byte[] m_txt = args[1].getBytes();
```

```
out.write(m_txt, 0, m_txt.length);
```

```
out.flush();
```

```
byte[] m_rcv = new byte[m_txt.length];
```

```
int n = in.read(m_rcv, 0, m_rcv.length);
```

```
if (n != m_rcv.length) {
```

```
    System.out.println("Some data are lost ...");
```

```
}
```

```
System.out.println(new String(m_rcv));
```

```
out.close();
```

```
in.close();
```

```
clientSocket.close();
```

```
}
```

```
}
```

## 21. simple server program

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

```
import java.net.*;
```

```
public class SimpleServer {
```

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

```
        boolean listening = true;
```

```
        ServerSocket serverSocket = null;
```

```
        try {
```

```
            serverSocket = new ServerSocket(4442);
```

```
        } catch (IOException e) {
```

```
            System.err.println("Could not listen on port: 4444.");
```

```
            System.exit(1);
```

```
        }
```

# JAVA PROGRAMMING LANGUAGE

---

```
while(listening) {  
    Socket clientSocket = serverSocket.accept();  
    (new SimpleConHandler(clientSocket)).start();  
}
```

```
serverSocket.close();  
}  
}
```

Connection Handler:

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

```
import java.net.*;
```

```
public class SimpleConHandler extends Thread
```

```
{
```

```
    private Socket clientSocket;
```

```
    public SimpleConHandler(Socket clientSocket) {
```

```
        this.clientSocket = clientSocket;
```

```
    }
```

```
    public void run() {
```

```
BufferedInputStream in;

BufferedOutputStream out;

try {

    in = new BufferedInputStream(dientSocket.getInputStream());
    out = new BufferedOutputStream(dientSocket.getOutputStream());

} catch (IOException e) {

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

    return;

}

try {

    byte[] msg = new byte[4096];

    int bytesRead = 0;

    int n;

    while((n = in.read(msg, bytesRead, 256)) != -1) {

        bytesRead += n;

        if (bytesRead == 4096) {

            break;

        }

        if (in.available() == 0) {
```

```
        break;
    }
}

    for(int i=bytesRead;i>0; i--) {
out.write(msg[i-1]);
    }

    out.flush();

} catch(IOException e1) {
    System.out.println(e1.toString());
}

try {
    out.close();
    in.close();
    clientSocket.close();
} catch ( IOException e2 ) {;}
}
}
```

## 22. count no.of lines,words,characters from given file

```
import java.util.*;

import java.lang.*;

import java.io.*;

class CountLWL

{

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

{

FileInputStream f=new FileInputStream("cherry.txt");

//Byte b[]=f.getBytes();

int wc=1,ln=1,i=0,k;

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

while((k=f.read())!=-1)

{

i++;

if(k==' ' || k==' ' || k=='\n')

wc=wc+1;

if(k=='\n')

ln=ln+1;

}

System.out.println("No of characters" +(i-ln));

System.out.println("No of words"+wc);
```

```
System.out.println("Noof lines"+ln);
```

```
}
```

```
}[note: create cherry.txt file at c drive or u r present directory]
```

### 23. change specific character in the given file

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

```
import java.lang.*;
```

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

```
class CharChange
```

```
{
```

```
public static void main(String arg[]) throws IOException
```

```
{
```

```
FileInputStream f=new FileInputStream("cher.txt");
```

```
FileOutputStream f1=new FileOutputStream("cher1.txt");
```

```
int i=0,k;
```

```
char target='s';
```

```
while((k=f.read())!=-1)
```

```
{
```

```
if(k=='c')
```

```
{
```

```
f1.write(target);
```

```
i++;
```

```
}
```

```
else
```

```
f1.write(k);
```

```
}  
  
System.out.println(i+"No of characters changed");  
  
}  
  
}
```

## 24. program for linear search

```
import java.io.*;  
  
class LinearSearch  
{  
  
    public static void main(String arg[]) throws IOException  
    {  
  
        int a[],key,n,i;  
  
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));  
  
        System.out.println("Enter array size");  
  
        n=Integer.parseInt(br.readLine());  
  
        a=new int[n];  
  
        System.out.println("Enter array values");  
  
        for(i=0;i<n;i++)  
        {  
  
            a[i]=Integer.parseInt(br.readLine());  
  
        }  
  
        System.out.println("Enter key value");  
  
        key=Integer.parseInt(br.readLine());  
  
        linear(a,key,n);  
  
    }  
}
```

```
static void linear(int a[],int key,int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        if(key==a[i])
        {
            System.out.println("Key Found");
            break;
        }
    }
    if(i>=n)
    System.out.println("key not found");
}}
```

## 25. program for binary search

```
import java.io.*;

class BinarySearch
{
    public static void main(String arg[]) throws IOException
    {
        int a[],key,n,i;

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Enter array size");

        n=Integer.parseInt(br.readLine());
```

```
a=new int[n];

System.out.println("Enter array values in ascending order");

for(i=0;i<n;i++)

{

a[i]=Integer.parseInt(br.readLine());

}

System.out.println("Enter key value");

key=Integer.parseInt(br.readLine());

binary(a,key,n);

}

static void binary(int a[],int key,int n)

{

int r=0;

int low=0;

int high=n;

int mid=low+high/2;

while((mid>=0) && (mid<n))

{

if(a[mid]<=key)

{

if(a[mid]==key)

{

r=1;

break;
```

```
}  
  
else  
  
mid=(mid+1+high)/2;  
  
}  
  
else if(a[mid]>=key)  
  
{  
  
if(a[mid]==key)  
  
{  
  
r=1;  
  
break;  
  
}  
  
else  
  
mid=(low+mid)/2;  
  
}  
  
}  
  
if(r==1)  
  
System.out.println("Key Found");  
  
else  
  
System.out.println("Key Not Found");  
  
}  
  
}
```

## 26. program for bubble sort

```
import java.io.*;

import java.util.*;

class Bubble

{

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

    {

        int a[],n;

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Enter array size");

        n=Integer.parseInt(br.readLine());

        a=new int[n];

        System.out.println("Enter values");

        for(int i=0;i<n;i++)

        {

            a[i]=Integer.parseInt(br.readLine());

        }

        bubble(a,n);

    }

    static void bubble(int a[],int n)

    {

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

```
{  
for(int j=0;j<n-1;j++)  
{  
if(a[j]>=a[j+1])  
{  
a[j]+=a[j+1];  
a[j+1]=a[j]-a[j+1];  
a[j]=a[j]-a[j+1];  
}  
}  
}  
  
System.out.println("Sorted List");  
  
for(int i=0;i<n;i++)  
System.out.println(a[i]);  
  
}
```

### **27. Program for insertion sort**

```
import java.io.*;  
  
import java.util.*;  
  
class Insertion  
{  
  
public static void main(String arg[]) throws IOException  
{  
  
int a[],n;
```

# JAVA PROGRAMMING LANGUAGE

---

```
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter array size");

n=Integer.parseInt(br.readLine());

a=new int[n];

System.out.println("Enter values");

for(int i=0;i<n;i++)

{

a[i]=Integer.parseInt(br.readLine());

}

insertion(a,n);

}


static void insertion(int a[],int n)

{

for(int i=1;i<n;i++)

{

int j=i;

int B=a[i];

while((j>0) && (a[j-1]>B))

{

a[j]=a[j-1];

j--;

}

a[j]=B;
```

```
}  
  
System.out.println("Sorted List");  
  
for(int i=0;i<n;i++)  
  
System.out.println(a[i]);  
  
}  
  
}
```

## 28. Program for selection sort

```
import java.io.*;  
  
import java.util.*;  
  
class Selection  
{  
  
    public static void main(String arg[]) throws IOException  
    {  
  
        int a[],n;  
  
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));  
  
        System.out.println("Enter array size");  
  
        n=Integer.parseInt(br.readLine());  
  
        a=new int[n];  
  
        System.out.println("Enter values");  
  
        for(int i=0;i<n;i++)  
  
        {  
  
            a[i]=Integer.parseInt(br.readLine());  
  
        }  
  
        selection(a,n);  

```

```
}

static void selection(int a[],int n)

{
    int i,j,first,temp;
    for(i=a.length-1;i>0;i--)
    {
        first=0;
        for(j=1;j<=i;j++)
        {
            if( a[j]>a[first] )
            first = j;
        }
        temp=a[first];
        a[first]=a[i];
        a[i]=temp;
    }

    System.out.println("Sorted List");

    for(i=0;i<n;i++)

        System.out.println(a[i]);

}
```

## 29. Program for Quick sort

```
import java.io.*;

import java.util.*;

class Quick

{

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

    {

        int a[],n;

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Enter array size");

        n=Integer.parseInt(br.readLine());

        a=new int[n];

        System.out.println("Enter values");

        for(int i=0;i<n;i++)

        {

            a[i]=Integer.parseInt(br.readLine());

        }

        quick(a,0,n-1);

        System.out.println("Sorted List");

        for(int i=0;i<n;i++)

            System.out.println(a[i]);

    }

}
```

```
static void quick(int a[],int left,int right)
{
    int l,r,pivot;
    l=left;
    r=right;
    pivot=a[left];
    while(left<right)
    {
        while((a[right]>=pivot) && (left<right))
            right--;
        if(left!=right)
        {
            a[left]=a[right];
            left++;
        }
        while((a[left]<=pivot) && (left<right))
            left++;
        if(left!=right)
        {
            a[right]=a[left];
            right--;
        }
    }
    a[left]=pivot;
```

```
    pivot=left;
    left=l;
    right=r;
    if(left<pivot)
        quick(a,left,pivot);
    if(right>pivot)
        quick(a,pivot+1,right);
    }
}
```

### 30. Implementation of stack ADT

```
class Node
{
    Node next;
    int data;
}

class Stack
{
    Node head;
    int tos;

    Stack()
    {
        tos=-1;
    }

    void push(int x)
```

```
{
    if(tos<0)
    {
        head=new Node();
        head.next=null;
        head.data=x;
        tos++;
    }
    else
    {
        Node temp,temp1;
        temp=head;
        if(tos>=4)
        {
            System.out.println("Stack Over Flow");
        }
        tos++;
        while(temp.next!=null)
            temp=temp.next;
        temp1=new Node();
        temp.next=temp1;
        temp1.next=null;
        temp1.data=x;
    }
}
```

## JAVA PROGRAMMING LANGUAGE

---

```
}  
  
void display()  
{  
    Node temp=head;  
    if(tos<0)  
    {  
        System.out.println("Stack under flow");  
    }  
    while(temp!=null)  
    {  
        System.out.println(temp.data+" ");  
        temp=temp.next;  
    }  
}  
  
void pop()  
{  
    Node temp=head;  
    Node temp1=new Node();  
    if(tos<0)  
    {  
        System.out.println("Stack Under Flow");  
    }  
    tos--;  
    while(temp.next!=null)
```

```
{  
temp1=temp;  
temp=temp.next;  
}  
temp1.next=null;  
}  
}  
class StackList  
{  
public static void main(String args[])  
{  
Stack s=new Stack();  
s.push(1);  
s.push(2);  
s.push(3);  
s.push(4);  
s.display();  
s.pop();  
s.display();  
}  
}
```

## 31. Implementation of queue ADT

```
class Node
{
    Node next;
    int data;
}

class Queue
{
    Node head;
    int front,rare;

    Queue()
    {
        front=-1;
        rare=-1;
    }

    void enqueue(int x)
    {
        if(rare<0)
        {
            head=new Node();
            head.next=null;
            head.data=x;
```

```
rare++;  
  
}  
  
else  
  
{  
  
Node temp,temp1;  
  
temp=head;  
  
if(rare>=4)  
  
{  
  
System.out.println("Queue Over Flow");  
  
}  
  
rare++;  
  
while(temp.next!=null)  
  
temp=temp.next;  
  
temp1=new Node();  
  
temp.next=temp1;  
  
temp1.next=null;  
  
temp1.data=x;  
  
}  
  
}  
  
void display()  
  
{  
  
Node temp=head;  
  
if(rare<0)  
  
{
```

```
System.out.println("Queue under flow");

}

while(temp!=null)

{

System.out.println(temp.data+" ");

temp=temp.next;

}

}

void dequeue()

{

Node temp=head;

Node temp1=new Node();

if(rare<0)

{

System.out.println("Queue Under Flow");

}

if(front==rare)

{

front=rare=-1;

head=null;

}

front++;

head=head.next;

}
```

```
}
```

```
class QueueList
```

```
{
```

```
public static void main(String args[])
```

```
{
```

```
Queue s=new Queue();
```

```
s.enqueue(1);
```

```
s.enqueue(2);
```

```
s.enqueue(3);
```

```
s.enqueue(4);
```

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

```
s.dequeue();
```

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

```
}
```

```
}
```

## **32. Example of Thread implementation of program**

```
class Account
```

```
{
```

```
int b=10;
```

```
public synchronized void deposit(int a)
```

```
{
```

```
b=b+a;
```

```
System.out.println("hi");
```

## JAVA PROGRAMMING LANGUAGE

---

```
}

public int getBal()

{

return b;

}

}

class Customer extends Thread

{

Account a;

public Customer(Account a)

{

this.a=a;

}

public void run()

{

a.deposit(10);

}

}

import java.lang.*;

class Bank

{

public static void main(String arg[])

{

int i;
```

# JAVA PROGRAMMING LANGUAGE

---

```
Account a=new Account();

Customer c[]=new Customer[10];

for( i=0;i<10;i++)

{

    c[i]=new Customer(a);

    c[i].start();


    try

    {

        c[i].join();

    }

    catch(InterruptedException e)

    {

        System.out.println(e);

    }

    System.out.println("bal="+a.getBal());

}

}

}
```

### 33. Example of Mouse event program

```
import java.io.*;

import java.util.*;

import java.awt.*;

import java.awt.event.*;

public class MouseEventsh extends Applet implements MouseListener, MouseMotionListener
{
    String m=" ";

    int x=10;

    int y=20;

    public void init()
    {
        addMouseListener(this);

        addMouseMotionListener(this);

    }

    public void mouseClicked(MouseEvent m)
    {
        x=m.getX();

        y=m.getY();

        m="mouse clicked";

        repaint();

    }
}
```

```
public void mouseReleased(MouseEvent m)
{
    x=m.getX();
    y=m.getY();
    m="mouse Released";
    repaint();
}public void mousePressed(MouseEvent m)
{
    x=m.getX();
    y=m.getY();
    m="mouse Released";
    repaint();
}public void mouseEntered(MouseEvent m)
{
    x=m.getX();
    y=m.getY();
    m="mouse Released";
    repaint();
}public void mouseExited(MouseEvent m)
{
    x=m.getX();
    y=m.getY();
    m="mouse Released";
    repaint();
```

```
}public void mouseDragged(MouseEvent m)
{
    x=m.getX();
    y=m.getY();
    m="mouse Released";
    repaint();
}

public void mouseMoved(MouseEvent m)
{
    x=m.getX();
    y=m.getY();
    m="mouse Released";
    repaint();
}

public void paint(Graphics g)
{
    g.drawString(m,x,y);
}
}
```

### **Html File for mouse events**

```
<html>

<head>

<body>

    <applet align="center" code="MouseEventSh" width="100" height="200">
```

</applet>

</body>

</head>

</html>

### **34. Write a program for textbox color change based on mouse events**

```
import java.awt.*;
```

```
import java.awt.event.*;
```

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

```
public class MouseInfo extends Frame implements MouseListener
```

```
{//Frame f=new Frame();
```

```
public MouseInfo()
```

```
{
```

```
setTitle("mouse events");
```

```
setSize(200,200);
```

```
setVisible(true);
```

```
addMouseListener(this);
```

```
}
```

```
public void mouseEntered(MouseEvent m)
```

```
{
```

```
setBackground(Color.red);
```

```
}
```

```
public void mouseExited(MouseEvent m)
```

```
{
```

## JAVA PROGRAMMING LANGUAGE

---

```
setBackground(Color.green);  
}  
  
public void mousePressed(MouseEvent m)  
{  
    setBackground(Color.blue);  
}  
  
public void mouseReleased(MouseEvent m)  
{  
    setBackground(Color.green);  
}  
  
public void mouseClicked(MouseEvent m)  
{  
    setBackground(Color.red);  
}  
  
public static void main(String arg[])  
{  
    new MouseInfo();  
}  
}
```

### **Html file:**

```
<html>  
  
<head>  
  
<body>  
  
<applet align="center" code="MouseInfo" width="100" height="200">
```

</applet>

</body>

</head>

</html>

## 35. Program for Key events

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

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

```
import java.lang.*;
```

```
import java.applet.*;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
public class KeyboardEvents extends Applet implements KeyListener
```

```
{
```

```
String m="key pressed ";
```

```
int x=10;
```

```
int y=20;
```

```
public void init()
```

```
{
```

```
addKeyListener(this);
```

```
requestFocus();
```

```
}
```

```
public void keyTyped(KeyEvent k)
```

```
{
```

```
m=m+k.getKeyChar();
```

```
repaint();  
  
}  
  
public void keyPressed(KeyEvent k)  
{  
  
    showStatus("Key pressed");  
  
}  
  
public void keyReleased(KeyEvent k)  
{  
  
    showStatus("key released");  
  
}  
  
public void paint(Graphics g)  
{  
  
    g.drawString(m,x,y);  
  
}  
  
}
```

**Html file:**

```
<html>  
  
<head>  
  
<body>  
  
    <applet align="center" code="KeyboardEvents" width="100" height="200">  
  
    </applet>  
  
</body>  
  
</head>  
  
</html>
```

## 36. Program for keyboard Listener

```
import java.util.*;

import java.io.*;

import java.lang.*;

import java.applet.*;

import java.awt.*;

import java.awt.event;

public class KeyboardEvents extends Applet implements KeyListener

{

String m=" ";

int x=10;

int y=20;

public void init()

{

addKeyListener(this);

requestFocus();

}

public void keyTyped(KeyEvent k)

{

m=m+k.getKeyChar();

repaint();

}
```

# JAVA PROGRAMMING LANGUAGE

---

```
public void keyPressed()
{
    showStatus("Key pressed");
}

public void keyReleased()
{
    showStatus("key released");
}

public void paint(Graphics g)
{
    g.drawString(m,x,y);
}
}
```

## **Html file:**

```
<html>

<head>

<body>

<applet align="center" code="KeyboardEvents" width="100" height="200">

</applet>

</body>

</head>

</html>
```

## 37. Program for Creating a JTABLE

```
import javax.swing.*;

class JButtonDemo

{

public static void main(String arg[])

{

new JButtonDemo();

}JButtonDemo()

{

JFrame f=new JFrame("title");

JPanel p=new JPanel();

JButton b=new JButton("press me");

p.add(b);

f.setContentPane(p);

f.show();

}

}
```

### Html file:

```
<html>

<head>

<body>

<applet align="center" code="Jtable" width="100" height="200">
```

</applet>

</body>

</head>

</html>

## **38. Program for Creating JButton**

```
import javax.swing.*;

class JButtonDemo

{

public static void main(String arg[])

{

new JButtonDemo();

}JButtonDemo()

{

JFrame f=new JFrame("title");

JPanel p=new JPanel();

JButton b=new JButton("press me");

p.add(b);

f.setContentPane(p);

f.show();

}

}
```

## 39. Program for Creating rectangle , square etc..

```
import java.applet.*;

import java.awt.*;

public class GraphicsDemo extends Applet

{

    public void paint(Graphics g)

    {

        String s="cherry";

        g.fillRect(20,20,20,30);

        g.drawRect(35,35,30,40);

        g.fillRoundRect(100,100,10,20,20,20);

        g.drawRoundRect(120,120,10,20,20,20);

        g.fillOval(130,130,10,20);

        g.drawOval(150,150,20,30);

        g.fillArc(100,40,70,10,75,80);

        g.drawArc(100,40,70,10,75,80);

        g.drawLine(0,0,250,250);

        g.drawString(s,10,15);

    }

}
```

### HTML FILE:

```
<html>
```

```
<head>

<body>

<applet align="center" code="GraphicsDemo" width="100" height="200">

</applet>

</body>

</head>

</html>
```

#### 40. Program for implement a simple calculator

```
import java.awt.*;

import java.awt.event.*;

import javax.swing.*;

import javax.swing.event.*;


class Calculator extends JFrame {

    private final Font BIGGER_FONT = new Font("monospaced",

    Font.PLAIN, 20);

    private JTextField textfield;

    private boolean number = true;

    private String equalOp = "=";

    private CalculatorOp op = new CalculatorOp();


    public Calculator() {

        textfield = new JTextField("0", 12);

        textfield.setHorizontalAlignment(JTextField.RIGHT);
```

# JAVA PROGRAMMING LANGUAGE

---

```
textfield.setFont(BIGGER_FONT);

ActionListener numberListener = new NumberListener();

String buttonOrder = "1234567890 ";

JPanel buttonPanel = new JPanel();

buttonPanel.setLayout(new GridLayout(4, 4, 4, 4));

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

    String key = buttonOrder.substring(i, i+1);

    if (key.equals(" ")) {

        buttonPanel.add(new JLabel(""));

    } else {

        JButton button = new JButton(key);

        button.addActionListener(numberListener);

        button.setFont(BIGGER_FONT);

        buttonPanel.add(button);

    }

}

ActionListener operatorListener = new OperatorListener();

JPanel panel = new JPanel();

panel.setLayout(new GridLayout(4, 4, 4, 4));

String[] opOrder = {"+", "-", "*", "/", "=", "C"};

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

    JButton button = new JButton(opOrder[i]);

    button.addActionListener(operatorListener);
```

# JAVA PROGRAMMING LANGUAGE

---

```
button.setFont(BIGGER_FONT);

panel.add(button);

}

JPanel pan = new JPanel();

pan.setLayout(new BorderLayout(4, 4));

pan.add(textfield, BorderLayout.NORTH );

pan.add(buttonPanel , BorderLayout.CENTER);

pan.add(panel , BorderLayout.EAST );

this.setContentPane(pan);

this.pack();

this.setTitle("Calculator");

this.setResizable(false);

}

private void action() {

number = true;

textfield.setText("0");

equalOp = "=";

op.setTotal("0");

}

class OperatorListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

if (number) {

action();

textfield.setText("0");
```

```
} else {  
  
    number = true;  
  
    String displayText = textfield.getText();  
  
    if (equalOp.equals("=")) {  
  
        op.setTotal(displayText);  
  
    } else if (equalOp.equals("+")) {  
  
        op.add(displayText);  
  
    } else if (equalOp.equals("-")) {  
  
        op.subtract(displayText);  
  
    } else if (equalOp.equals("*")) {  
  
        op.multiply(displayText);  
  
    } else if (equalOp.equals("/")) {  
  
        op.divide(displayText);  
  
    }  
  
    textfield.setText("" + op.getTotalString());  
  
    equalOp = e.getActionCommand();  
  
    }  
  
    }  
  
    }  
  
    class NumberListener implements ActionListener {  
  
        public void actionPerformed(ActionEvent event) {  
  
            String digit = event.getActionCommand();  
  
            if (number) {  
  
                textfield.setText(digit);
```

```
        number = false;

    } else {

textfield.setText(textfield.getText() + digit);

    }

    }

    }

    public class CalculatorOp {

private int total;

    public CalculatorOp() {

        total = 0;

    }

    public String getTotalString() {

        return ""+total;

    }

    public void setTotal(String n) {

        total = convertToNumber(n);

    }

    public void add(String n) {

        total += convertToNumber(n);

    }

    public void subtract(String n) {

        total -= convertToNumber(n);

    }

    }
```

```
public void multiply(String n) {  
    total *= convertToNumber(n);  
}  
  
public void divide(String n) {  
    total /= convertToNumber(n);  
}  
  
private int convertToNumber(String n) {  
    return Integer.parseInt(n);  
}  
}  
}
```

**Another class which name Swing calculator .java :**

```
import java.awt.*;  
import java.awt.event.*;  
import javax.swing.*;  
import javax.swing.event.*;  
  
class SwingCalculator {  
    public static void main(String[] args) {  
        JFrame frame = new Calculator();  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setVisible(true);  
    }  
}
```

```
}
```

**Html file:**

```
<HTML>
```

```
<HEAD>
```

```
</HEAD>
```

```
<BODY>
```

```
<APPLET ALIGN="CENTER" CODE="Calculator" WIDTH="400" HEIGHT="200"></APPLET>
```

```
</BODY>
```

```
</HTML>
```

## **41. Create simple form using all AWT components.**

```
import java.awt.*;
```

```
import java.applet.Applet;
```

```
public class Components extends Applet {
```

```
    public void init() {
```

```
        Frame aFrame = new Frame("Button Test");
```

```
        setLayout(new GridLayout(3,2));
```

```
        Label aLabel = new Label("Enter Email id:");
```

```
        aLabel.setAlignment(Label.LEFT);
```

```
        aLabel.setBackground(Color.red);
```

```
        add(aLabel);
```

```
        TextField emailTextField = new TextField(30);
```

```
        add(emailTextField);
```

# JAVA PROGRAMMING LANGUAGE

---

```
Label aLabel1 = new Label("Enter password:");
aLabel1.setAlignment(Label.LEFT);
aLabel1.setBackground(Color.green);
add(aLabel1);

TextField passwordTextField = new TextField(30);
passwordTextField.setEchoChar('*');
add(passwordTextField);

Label aLabel2 = new Label("Enter sex:");
aLabel2.setAlignment(Label.LEFT);
aLabel2.setBackground(Color.blue);
add(aLabel2);

Choice aChoice = new Choice();
aChoice.add("female");
aChoice.add("male");

String selectedDestination= aChoice.getSelectedItem();
add(aChoice);

setLayout(new GridLayout(10,1));

Label aLabel3 = new Label("select your fav colors:");
aLabel3.setAlignment(Label.LEFT);
aLabel3.setBackground(Color.red);
add(aLabel3);

Checkbox c = new Checkbox("red");

Checkbox c1 = new Checkbox("blue");

Checkbox c2 = new Checkbox("green");
```

## JAVA PROGRAMMING LANGUAGE

---

```
Checkbox c3 = new Checkbox("black");

Checkbox c4= new Checkbox("white");

add(c);

add(c1);

add(c2);

add(c3);

add(c4);

Label aLabel5 = new Label("select your country from below list:");

aLabel5.setAlignment(Label.LEFT);

aLabel5.setBackground(Color.green);

add(aLabel5);

List aList = new List();

aList.add("INDIA");

aList.add("USA");

aList.add("LONDON");

aList.add("AUSTRALIA");

aList.setMultipleMode(false);

add(aList);

Label aLabel4 = new Label("enter your comments here:");

aLabel4.setAlignment(Label.LEFT);

aLabel4.setBackground(Color.blue);

add(aLabel4);

TextArea fullAddressTextArea = new TextArea(5, 80);

add(fullAddressTextArea);
```

# JAVA PROGRAMMING LANGUAGE

---

```
Button okButton = new Button("Ok");

Button cancelButton = new Button("Cancel");

okButton.setBackground(Color.red);

cancelButton.setBackground(Color.green);

add(okButton);

add(cancelButton);

}
```

```
}
```

## **HTML FILE:**

```
<HTML>
```

```
<HEAD>
```

```
</HEAD>
```

```
<BODY>
```

```
<APPLET ALIGN="CENTER" CODE="Components" WIDTH="400" HEIGHT="200"></APPLET>
```

```
</BODY>
```

```
</HTML>
```

## 42. Program for all swing components creation.

```
import javax.swing.*;

class JButtonDemo

{

public static void main(String arg[])

{

new JButtonDemo();

}

JButtonDemo()

{

String course[] = {"BCA", "MCA", "PPC", "CIC"};

JFrame f=new JFrame("title");

JPanel p=new JPanel();

f.setContentPane(p);

f.show();

JComboBox combo = new JComboBox(course);

//combo.setBackground(Color.gray);

//combo.setForeground(Color.red);

p.add(combo);

JLabel jl = new JLabel("ENTER YOUR NAME");

p.add(jl);

jl.setSize(100, 100); // pack();
```

# JAVA PROGRAMMING LANGUAGE

---

```
jlb.setVisible(true);

JTextField t1 = new JTextField(30);

f.add(t1);

JLabel jlb1 = new JLabel("select your fav colors");

p.add(jlb1);

JCheckBox cb1 = new JCheckBox("red"), cb2 = new JCheckBox(
    "blue"), cb3 = new JCheckBox("green");

p.add(cb1);

p.add(cb2);

p.add(cb3);

JLabel jlb2 = new JLabel("select sex");

p.add(jlb2);

JRadioButton aButton = new JRadioButton("female",true);

JRadioButton bButton = new JRadioButton("male");

p.add(aButton);

p.add(bButton);

JButton b=new JButton("press me");

p.add(b);

}

}
```

## 43. Program for Creating scroll bar

```
import javax.swing.*;

public class JScrollBarDemo

{

public static void main(String arg[])

{

new JScrollBarDemo();

}

JScrollBarDemo()

{

JScrollBar jsb=new JScrollBar();

add(jsb);

this.setSize(100,100);

setVisible(true);

}

}

}
```

## 44. Program for Creating menu bar

```
import javax.swing.*;

import java.awt.*;

public class JMenuBarDemo

{

    public static void main(String arg[])

    {

        new JMenuBarDemo();

    }

    JMenuBarDemo()

    {

        JMenuBar jmb=new JMenuBar();

        add(jmb);

        this.setSize(100,100);

        setVisible(true);

    }

}
```

## 45. Program for Creating Choice Box

```
import javax.swing.*;

import java.awt.*;

class JChoiceDemo

{

    public static void main(String arg[])

    {

        new JChoiceDemo();

    }

    JChoiceDemo()

    {

        JFrame f=new JFrame("title");

        JPanel p=new JPanel();

        JChoiceDemo c=new JChoiceDemo();

        c.add("catogry");

        String selectedDestination=c.getSelectedItem();

        p.add(c);

        f.setContentPane(p);

        f.show();

    }

}
```

## 46. Program for Creating List Box

```
import javax.swing.*;

import java.awt.*;

class JListDemo

{

    public static void main(String arg[])

    {

        new JListDemo();

    }

    JListDemo()

    {

        JFrame f=new JFrame("title");

        JPanel p=new JPanel();

        JList l=new JList();

        l.add("catgory");

        l.add("Edmonton");

        l.add("region");

        l.setMultipleMode(true);

        p.add(l);

        f.setContentPane(p);

        f.show();

    }

}
```

```
}
```

## 47. Program for Creating List Box

```
import javax.swing.*;

class JCheckBoxDemo1

{

public static void main(String arg[])

{

new JCheckBoxDemo1();

}

JCheckBoxDemo1()

{

JFrame f=new JFrame("title");

JPanel p=new JPanel();

JCheckBox creamcb=new JCheckBox("cream");

JCheckBox sugarcb=new JCheckBox("sugar");

if(creamcb.getState())

{

coffee.addCream();

}

p.add(creamcb);

p.add(sugarcb);

f.setContentPane(p);

f.show();
```

}

}