

Lab List

- 1 .Design and develop a java program to implement the types of arrays.
2. Design and develop a java program to demonstrate types of constructors.
3. Design and develop a java program for implementation of method overloading
4. Write a java program to implement the concept of packages.
5. Design and develop a program in java to implement the concept of Exception Handling
6. Design and develop a program in java to demonstrate the concept of multithreading.
7. Write a menu based program in java that occurs a shopping list of items from command line and store in a vector & perform operations.
8. Develop a program in java to perform string operations.
9. Develop a program in java to display calendar data by using the java utilities.
10. Write a java program to create an applet to draw a bar chart from the following details.

Subjects	Telugu	English	Maths	Physics	
Marks	60	65	70	85	90

11. Develop a program in java to implement the concept of Swings.
12. Develop a program in java to retrieve IP address from the system.
13. Design and develop a java program to passing message to client and server.
14. Write a program in java to connect data base created in MS-access using JDBC concept and perform operations.

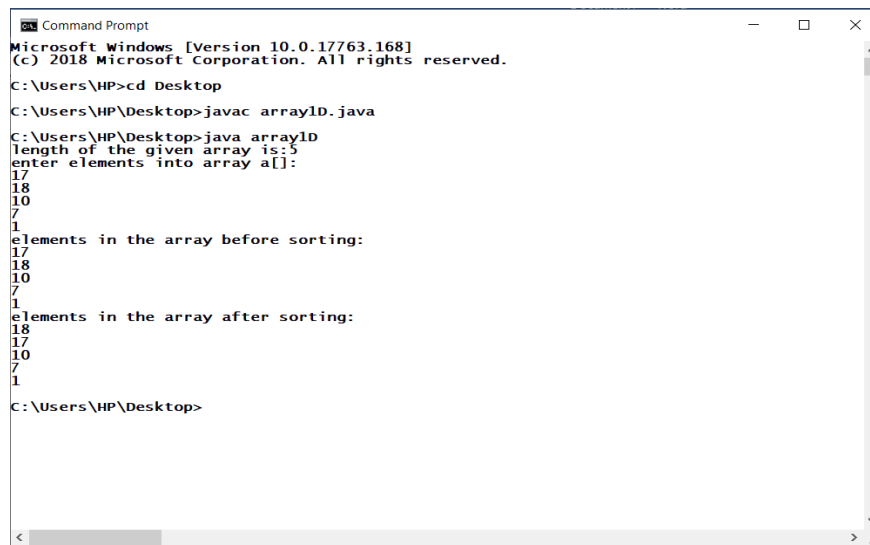
1 .Design and develop a java program to implement the types of arrays.

```
import java.util.*;

class array1D
{
    public static void main(String args[])
    {
        int a[]=new int[5];
        //int b[];
        //b=a;
        int i,j;
        int n=a.length;
        Scanner sc=new Scanner(System.in);
        System.out.println("length of the given array is:"+n);
        System.out.println("enter elements into array a[:]");
        for(i=0;i<n;i++)
        {
            a[i]=sc.nextInt();
        }
        System.out.println("elements in the array before sorting:");
        for(i=0;i<n;i++)
        {
            System.out.println(a[i]);
        }
        for(i=0;i<n;i++)
        {
            for(j=i+1;j<n;j++)
            {
                if(a[i]<a[j])
                {
                    int temp=a[i];
                    a[i]=a[j];
                    a[j]=temp;
                }
            }
        }
    }
}
```

```
}  
  
System.out.println("elements in the array after sorting:");  
for(i=0;i<n;i++)  
{  
    System.out.println(a[i]);  
}  
}  
}
```

Output:



```
Command Prompt  
Microsoft Windows [Version 10.0.17763.168]  
(c) 2018 Microsoft Corporation. All rights reserved.  
C:\Users\HP>cd Desktop  
C:\Users\HP\Desktop>javac array1D.java  
C:\Users\HP\Desktop>java array1D  
length of the given array is:5  
enter elements into array a[]:  
17  
18  
10  
7  
1  
elements in the array before sorting:  
17  
18  
10  
7  
1  
elements in the array after sorting:  
1  
7  
10  
17  
18  
C:\Users\HP\Desktop>
```

2. Design and develop a java program to demonstrate types of constructors.

```
import java.util.*;

class area
{
    Scanner sc=new Scanner(System.in);

    area()
    {
        System.out.println("\nEnter the radius of the circle");
        int r=sc.nextInt();

        System.out.println("\nThe area of the circle is :
        "+(3.14*r*r));

    }

    area(int i,int j)
    {
        System.out.println("\nThe area of the triangle is : "+(i*j/2));

    }

    area(float g)
    {
        System.out.println("\nThe area of the sphere is : "+(3.14*6*g*g));

    }

    area(float u,float v)
    {
        System.out.println("\nThe area of the rectangle is : "+(u*v));

    }

}

class java2
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);

        int ch;
```

```

do
{
System.out.println("\nTypesofconstructors\n1.Circle\n2.Triangle\n3.Sphere\n4.Rectangle\n");
System.out.println("\nEnter your choice");
ch=sc.nextInt();
switch(ch)
{
case 1:area a1=new area();
        break;

case 2:System.out.println("\nEnter the height and breadth of triangle");
        int h=sc.nextInt();
        int b=sc.nextInt();
        area a2=new area(h,b);
        break;

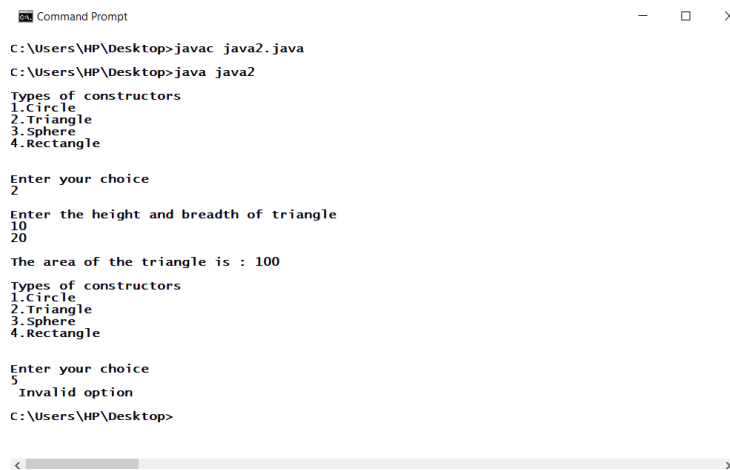
case 3:System.out.println("\nEnter the radius of the sphere");
        int s=sc.nextInt();
        area a3=new area(s);
        break;

case 4:System.out.println("\nEnter the length and breadth of triangle");
        int l=sc.nextInt();
        int br=sc.nextInt();
        area a4=new area(l,br);
        break;

default:System.out.println(" Invalid option");
}
}while(ch<5);
}
}

```

Output:



```
Command Prompt

C:\Users\HP\Desktop>javac java2.java
C:\Users\HP\Desktop>java java2
Types of constructors
1.Circle
2.Triangle
3.Sphere
4.Rectangle

Enter your choice
2
Enter the height and breadth of triangle
10
20
The area of the triangle is : 100
Types of constructors
1.Circle
2.Triangle
3.Sphere
4.Rectangle

Enter your choice
5
Invalid option
C:\Users\HP\Desktop>
```

3. **Design and develop a java program for implementation of method overloading**

```
class OverloadDemo
```

```
{
```

```
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: " + a + " " + b);
```

```
}
```

```
double test(double a)
```

```
{
```

```
System.out.println("double a: " + a);
```

```
return a*a;
```

```
}
```

```
}
```

```
class Overload
```

```
{
```

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

```
{
```

```
OverloadDemo ob = new OverloadDemo();
```

```
double result;
```

```
ob.test();
```

```
ob.test(10);
```

```
ob.test(10, 20);
```

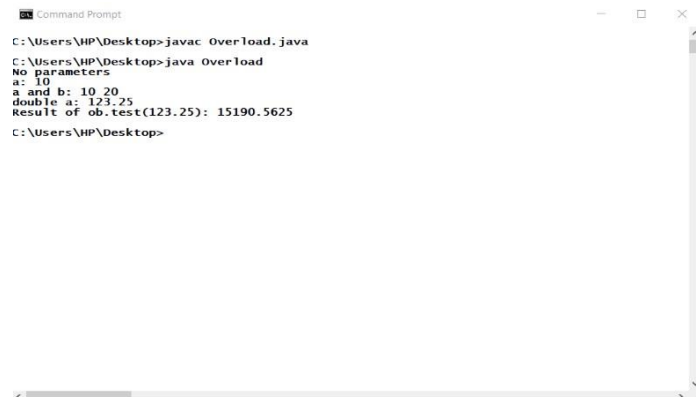
```
result = ob.test(123.25);
```

```
System.out.println("Result of ob.test(123.25): " + resu
```

```
}
```

}

Output:



```
Command Prompt
C:\Users\HP\Desktop>javac Overload.java
C:\Users\HP\Desktop>java Overload
No parameters
a: 10
a and b: 10 20
double a: 123.25
Result of ob.test(123.25): 15190.5625
C:\Users\HP\Desktop>
```

4. Write a java program to implement the concept of packages.

```
package book;

public class BookDetails

{
String name,author;
float price;
int year;
public BookDetails(String n,String a,float p,int y)
{
    name=n;
    author=a;
    price=p;
    year=y;
}

public void display()
{
    System.out.println("Book Name:"+name);
    System.out.println("Book Author:"+author);
    System.out.println("Book Price:Rs "+price);
    System.out.println("Year of Publishing:"+year);
}
}

import book.*;
class BookDemo
{
public static void main(String args[])
{
```

```
BookDetails b=new BookDetails("Java Programming", "Balguruswamy",  
190.00f,2007);  
b.display();  
}  
}
```

Output:

```
Book Name      :Java Programming  
Book Author    :Balguruswamy  
Book Price     :Rs 190.0  
Year of Publishing:2007
```

5. Design and develop a program in java to implement the concept of Exception Handling

```
import java.io.*;
import java.lang.*;
class MyException extends Exception
{
    String msg;
    public MyException(String message)
    {
        msg=message;
    }
    public String toString()
    {
        return msg;
    }
}
class exception9
{
    static int eno[]=new int[5];
    static String ename[]=new String[3];
    static String job[]=new String[3];
    static int sal[]=new int[3];
```

```

static int count=0;

static void get() throws IOException
{
    int i;

    DataInputStream d=new DataInputStream(System.in);

    try
    {
        for(i=count;i<3;i++)
        {
            System.out.println("\n enter"+(i+1)+"employee detail");
            System.out.println("\n enter employee no. and name:");
            eno[i]=Integer.parseInt(d.readLine());
            if(eno[i]<=0)
                throw new MyException("entr valid no.");
            ename[i]=d.readLine();
            System.out.println("\n enter job and salary");
            job[i]=d.readLine();
            sal[i]=Integer.parseInt(d.readLine());
            if(sal[i]<0)
                throw new MyException("enter valid salary");
            count++;
        }
    }
    catch(MyException e)
    {
        System.out.println("exception caught:"+e);
        get();
    }
}

static void disp()

```

```

{
for(int i=0;i<3;i++)
{
System.out.println(eno[i]+"\\t\\t"+ename[i]+"\\t\\t"+job[i]+"\\t\\t"+sal[i]);
}
}

public static void main(String args[]) throws IOException
{
get();
System.out.println("\\n eno \\t\\t ename \\t\\t job \\t\\tsalary\\n");
disp();
}
}

```

Output:

```

Enter 1 employee details:
Enter employee no and name : 5554
Exception caught : enter valid number
Enter 1 employee detail :
Enter employee no and name : 3001
R.Haricharan
Enter job and salary : lecturer 25000
Enter 2 employee detail :
Enter employee no and name : 5001
S.Kalaiselvan
Enter job and salary : reader
20000
Enter 3 employee detail :
Enter employee no and name : 7801
S.Rajadurai
Enter job and salary : director
23000
35
ENO   ENAME      JOB           SALARY
3001  R.Haricharan lecturer      25000
500   S.Kaliselvan reader        20000
7801  S.Rajadurai  director      23000

```

6. Design and develop a program in java to demonstrate the concept of multithreading.

```
class EvenThread extends Thread
{
    EvenThread(String name)
    {
        super(name);
    }
    public void run()
    {
        try
        {
            for(int i=1;i<15;i++)
            {
                if (i%2==0) System.out.println(getName()+"\t"+i);
                Thread.sleep(1000);
            }
            System.out.println("Exit from thread "+getName());
        }
        catch(InterruptedException e)
        {
            System.out.println("Thread "+getName()+" Interrupted");
        }
    }
}

class OddThread extends Thread
{
    OddThread(String name)
    {
        super(name);
    }
}
```

```
public void run()
{
try
{
for(int i=1;i<15;i++)
{
if (i%2!=0)
System.out.println(getName()+"\t"+i);
Thread.sleep(500);
}
System.out.println("Exit from thread "+getName());
}
catch(InterruptedException e)
{
System.out.println("Thread "+getName()+" Interrupted");
}
}
}

class ThreadDemo
{
public static void main(String args[])
{
EvenThread e=new EvenThread("Even");
OddThread o=new OddThread("Odd");
e.start();
o.start();
}
}
```

Output:

```
C:\Users\System8\Desktop>javac OddThread.java
C:\Users\System8\Desktop>java ThreadDemo
Odd      1
Even     2
Odd      3
Odd      5
Even     4
Odd      7
Odd      9
Even     6
Odd     11
Odd     13
Even     8
Exit from thread Odd
Even    10
Even    12
Even    14
Exit from thread Even
C:\Users\System8\Desktop>^U^U_
```

7. Write a menu based program in java that occurs a shopping list of items from command line and store in a vector & perform operations.

```
import java.util.*;

public class VectorDemo {

    public static void main(String args[]) {
        Vector v = new Vector(3, 2);
        System.out.println("Initial size: " + v.size());
        System.out.println("Initial capacity: " + v.capacity());

        v.addElement(new Integer(1));
        v.addElement(new Integer(2));
        v.addElement(new Integer(3));
        v.addElement(new Integer(4));
        System.out.println("Capacity after four additions: " + v.capacity());

        v.addElement(new Double(5.45));
        System.out.println("Current capacity: " + v.capacity());
    }
}
```

```

v.addElement(new Double(6.08));
v.addElement(new Integer(7));
System.out.println("Current capacity: " + v.capacity());

v.addElement(new Float(9.4));
v.addElement(new Integer(10));
System.out.println("Current capacity: " + v.capacity());

v.addElement(new Integer(11));
v.addElement(new Integer(12));
System.out.println("First element: " + (Integer)v.firstElement());
System.out.println("Last element: " + (Integer)v.lastElement());

if(v.contains(new Integer(3)))
    System.out.println("Vector contains 3.");
    Enumeration vEnum = v.elements();
System.out.println("\nElements in vector:");

while(vEnum.hasMoreElements())
    System.out.print(vEnum.nextElement() + " ");
System.out.println();
}
}

```

Output:

```
Command Prompt
Microsoft Windows [Version 10.0.17763.168]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\HP>cd Downloads
C:\Users\HP\Downloads>cd "java rem prog"
C:\Users\HP\Downloads\java rem prog>javac VectorDemo.java
Note: VectorDemo.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Note: VectorDemo.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
C:\Users\HP\Downloads\java rem prog>java VectorDemo
Initial size: 0
Initial capacity: 3
Capacity after four additions: 5
Current capacity: 5
Current capacity: 7
Current capacity: 9
First element: 1
Last element: 12
Vector contains 3.

Elements in vector:
1 2 3 4 5.45 6.08 7 9.4 10 11 12
C:\Users\HP\Downloads\java rem prog>_
```

8. Develop a program in java to perform string operations.

```
import java.io.*;
import java.util.*;
import java.lang.*;
class stringop
{
Scanner sc=new Scanner(System.in);
void con()
{
String a;
String b;
System.out.println("Enter First String");
a=sc.next();
System.out.println("Enter Second String");
b=sc.next();
String c=a+b;
System.out.println("\nUsing + Operator : "+c);
c=a.concat(b);
System.out.println("\n Using concatination method : "+c);
}
void comp()
{
String a;
String b;
System.out.println("Enter First String");
a=sc.next();
System.out.println("Enter Second String");
b=sc.next();
if(a.equals(b))
{
System.out.println("\nBoth strings are exactly equal");
}
else
{

```

```

System.out.println("\nBoth strings are exactly not equal");
}
}
void icomp()
{
String a;
String b;
System.out.println("Enter First String");
a=sc.next();
System.out.println("Enter Second String");
b=sc.next();
if(a.equalsIgnoreCase(b))
{
System.out.println("\nBoth strings are equal");
}
else
{
System.out.println("\nBoth strings are not equal");
}
}
void upper()
{
String a;
System.out.println("\nEnter a string in lower case");
a=sc.next();
System.out.println("\nOriginal String is : "+a);
System.out.println("\nThe String in upper case : "+a.toUpperCase());
}
void lower()
{
String a;
System.out.println("\nEnter a string in upper case");
a=sc.next();
System.out.println("\n Original string is : "+a);
System.out.println("\n String in to lower case : "+a.toLowerCase());
}
void len()
{
String a;
String b;
System.out.println("Enter First String");
a=sc.next();
System.out.println("Enter Second String");
b=sc.next();
System.out.println("\n The length of the second string is : "+a.length());
}

```

```

System.out.println("\nThe length of the second string is : "+b.length());
}
void sub()
{
String a;
a="S.V.University,Tirupati";
System.out.println(a.substring(4,14));
System.out.println(a.substring(4));
}
}
class java8
{
public static void main(String args[])
{
int ch;
stringop s=new stringop();
Scanner sc=new Scanner(System.in);
do
{
System.out.println("\nString Operations\n1.String Concatination\n2.String
Compare\n3.String Compare with ignore case\n4.String to Upper case\n5.String to lower
case\n6.String length\n7.String sub string\n8.Exit\n");
System.out.println("\nEnter your option");
ch=sc.nextInt();
switch(ch)
{
case 1:s.con();
break;
case 2:s.comp();
break;
case 3:s.icomp();
break;
case 4:s.upper();
break;
case 5:s.lower();
break;
case 6:s.len();
break;
case 7:s.sub();
break;
default: System.out.println("\n***** Invalid *****");
}
}while(ch<8);
}
}

```

Output:

String Operations

- 1.String Concatination
- 2.String Compare
- 3.String Compare with ignore case
- 4.String to Upper case
- 5.String to lower case
- 6.String length
- 7.String sub string
- 8.Exit

Enter your option

1

Enter First String

raja

Enter Second String

chandu

Using + Operator : rajachandu

Using concatination method : rajachandu

String Operations

- 1.String Concatination
- 2.String Compare
- 3.String Compare with ignore case
- 4.String to Upper case
- 5.String to lower case
- 6.String length
- 7.String sub string
- 8.Exit

Enter your option

2

Enter First String

raja

Enter Second String

RAJA

Both strings are exactly not equal

String Operations

- 1.String Concatination
- 2.String Compare
- 3.String Compare with ignore case
- 4.String to Upper case
- 5.String to lower case
- 6.String length
- 7.String sub string
- 8.Exit

Enter your option

3

Enter First String

CHANDU

Enter Second String

chandu

Both strings are equal

String Operations

1.String Concatination

2.String Compare

3.String Compare with ignore case

4.String to Upper case

5.String to lower case

6.String length

7.String sub string

8.Exit

Enter your option

4

Enter a string in lower case

raja

Original String is : raja

The String in upper case : RAJA

String Operations

1.String Concatination

2.String Compare

3.String Compare with ignore case

4.String to Upper case

5.String to lower case

6.String length

7.String sub string

8.Exit

Enter your option

5

Enetr a string in upper case

CHANDU

Original string is : CHANDU

String in to lower case : chandu

String Operations

1.String Concatination

2.String Compare

3.String Compare with ignore case

4.String to Upper case

5.String to lower case

6.String length

7.String sub string

8.Exit

Enter your option

6

Enter First String

rajachandu

Enter Second String

chandra

The length of the second string is : 10

The length of the second string is : 7

String Operations

1.String Concatination

2.String Compare

3.String Compare with ignore case

4.String to Upper case

5.String to lower case

6.String length

7.String sub string

8.Exit

Enter your option

7

University

University,Tirupati

String Operations

1.String Concatination

2.String Compare

3.String Compare with ignore case

4.String to Upper case

5.String to lower case

6.String length

7.String sub string

8.Exit

Enter your option

8

***** Invalid *****

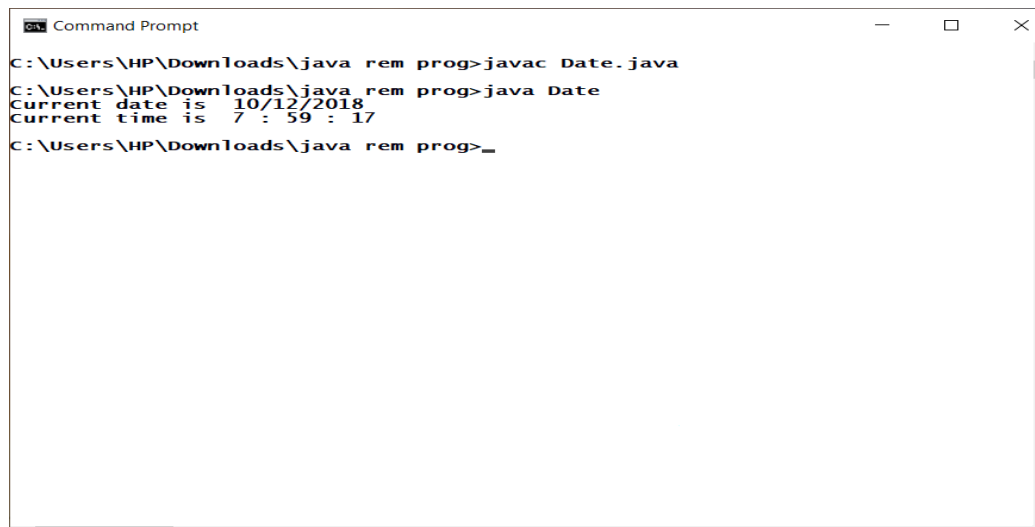
9. Develop a program in java to display calendar data by using the java utilities.

```
import java.util.*;

class Date
{
    public static void main(String args[])
    {
        int day, month, year;
        int second, minute, hour;
        GregorianCalendar date = new GregorianCalendar();
```

```
    day = date.get(Calendar.DAY_OF_MONTH);
    month = date.get(Calendar.MONTH);
    year = date.get(Calendar.YEAR);
    second = date.get(Calendar.SECOND);
    minute = date.get(Calendar.MINUTE);
    hour = date.get(Calendar.HOUR);
    System.out.println("Current date is "+day+"/"+(month+1)+"/"+year);
    System.out.println("Current time is "+hour+" : "+minute+" : "+second);
}
}
```

Output:



```
Command Prompt
C:\Users\HP\Downloads\java rem prog>javac Date.java
C:\Users\HP\Downloads\java rem prog>java Date
Current date is 10/12/2018
Current time is 7 : 59 : 17
C:\Users\HP\Downloads\java rem prog>_
```

10. Write a java program to create an applet to draw a bar chat from the following details.

Subjects	OR	JAVA	OS	DS	DBMS
Marks	60	65	70	85	90

```
import java.applet.*;
import java.awt.*;
public class BarChart extends Applet
{
int n=0;
String label[];
int value[];
```

```

public void init()
{
try
{
n= Integer.parseInt(getParameter("columns"));
label= new String[n];
value= new int[n];
label[0]=getParameter("label1");
label[1]=getParameter("label2");
label[2]= getParameter("label3");
label[3]= getParameter("label4");
value[0]=Integer.parseInt(getParameter("m1"));
value[1]=Integer.parseInt(getParameter("m2"));
value[2]=Integer.parseInt(getParameter("m3"));
value[3]= Integer.parseInt(getParameter("m4"));
}
catch(NumberFormatException e){ }
}

public void paint(Graphics g)
{
String msg=label[0]; for(int i=0;i<n;i++)
{
g.setColor(Color.red);
g.drawLine(100,50,100,350);
g.drawLine(100,350,300,350);
g.setColor(Color.blue);
g.drawString(label[i],40,145+(50*i));
g.setColor(Color.green);
g.fillRect(100,125+(50*i),value[i] ,30); showStatus("Bar chart Applet");
}
}
}

```

```
/*<APPLET      CODE="BarChart.class"      HEIGHT=400      WIDTH=500><PARAM
name="columns" VALUE=4>

<PARAM name="label1" VALUE="Tamil"><PARAM name="label2" VALUE="English">

<PARAM name="label3"      VALUE="Maths">

<PARAM name="label4"      VALUE="Physics"

<PARAM name="m1"          VALUE="78">

<PARAM name="m2"          VALUE="85">

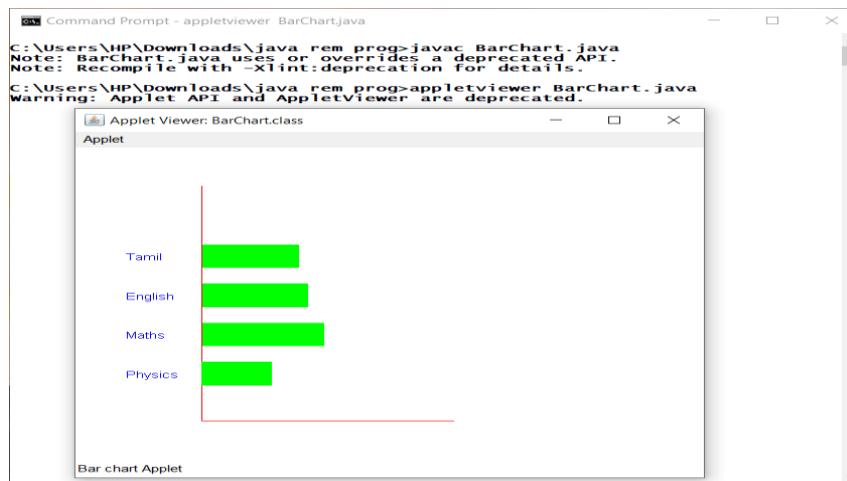
<PARAM name="m3"          VALUE="98"

<PARAM name="m4"          VALUE="56"

</APPLET>

*/
```

Output:



11. Develop a program in java to implement the concept of Swings.

```
import javax.swing.*;

public class JTabbedPaneDemo extends JApplet {
    public void init() {
        JTabbedPane jtp = new JTabbedPane();
        jtp.addTab("Cities", new CitiesPanel());
        jtp.addTab("Colors", new ColorsPanel());
        jtp.addTab("Flavors", new FlavorsPanel());
        getContentPane().add(jtp);
    }
}

class CitiesPanel extends JPanel {
    public CitiesPanel() {
        JButton b1 = new JButton("New York");
        add(b1);
        JButton b2 = new JButton("London");
        add(b2);
        JButton b3 = new JButton("Hong Kong");
        add(b3);
        JButton b4 = new JButton("Tokyo");
        add(b4);
    }
}

class ColorsPanel extends JPanel {
    public ColorsPanel() {
        JCheckBox cb1 = new JCheckBox("Red");
        add(cb1);
        JCheckBox cb2 = new JCheckBox("Green");
        add(cb2);
        JCheckBox cb3 = new JCheckBox("Blue");
        add(cb3);
    }
}
```

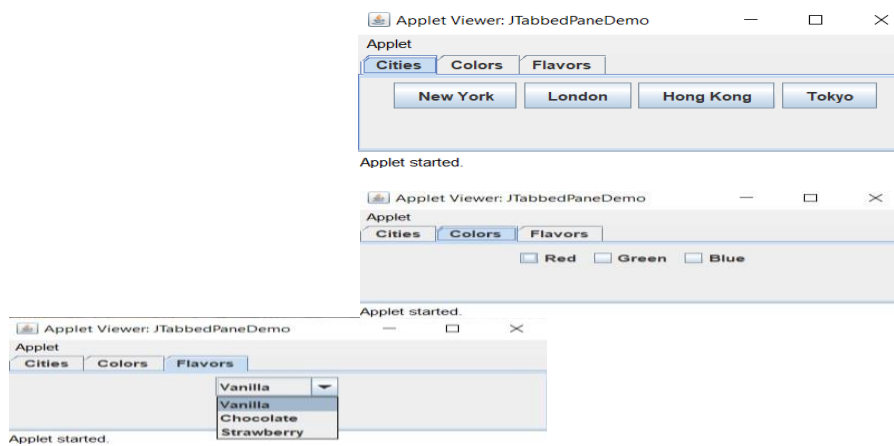
```

class FlavorsPanel extends JPanel {
public FlavorsPanel() {
JComboBox jcb = new JComboBox();
jcb.addItem("Vanilla");
jcb.addItem("Chocolate");
jcb.addItem("Strawberry");
add(jcb);
}
}

/*
<applet code="JTabbedPaneDemo" width=400 height=100>
</applet>
*/

```

Output:



12. Develop a program in java to retrieve IP address from the system.

```
import java.net.InetAddress;
import java.net.UnknownHostException;
public class MyIpAddress {
    public static void main(String args[]){
        try {
            InetAddress ipAddr = InetAddress.getLocalHost();
            System.out.println(ipAddr.getHostAddress());
        } catch (UnknownHostException ex) {
            ex.printStackTrace();
        }
    }
}
```

Output:



```
Command Prompt
C:\Users\HP\Desktop> javac MyIpAddress.java
C:\Users\HP\Desktop> java MyIpAddress
127.0.0.1
C:\Users\HP\Desktop>
```

13. Design and develop a java program to passing message to client and server.

FileClient.java

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

public class FileClient
{
    public static void main(String a[])
    {
        try
        {
            Socket con=new Socket("localhost",95);
            BufferedReader in=new BufferedReader(new InputStreamReader(System.in));
            DataOutputStream toserver=new DataOutputStream(con.getOutputStream());
            int size;
            System.out.print("Enter needed the file name:");
            String str=in.readLine();
            System.out.print("Enter the destination file name with extension:");
            String file=in.readLine(); toserver.writeBytes(str+"\n");
            DataInputStream fromserver=new DataInputStream(con.getInputStream());
            byte myarr[]=new byte[1024];
            int tot=0;
            FileOutputStream fout=new FileOutputStream(file);
            while((size=fromserver.read(myarr))>=0)
            {
                fout.write(myarr,0,size);
                tot=tot+size; System.out.println("Total size="+tot);
            }
        }
    }
}
```

```
fout.close();
System.out.println("Done ....Total bytes received:"+tot);
fromserver.close();
toserver.close();
con.close();
}
catch(UnknownHostException e)
{
System.out.println("Error in connection");
System.out.println(e.getMessage());
}
```

```

catch(IOException e)
{
System.out.println("Error in file transfer");
System.out.println(e.getMessage());
}
}
}

```

FileServer.java

```

import java.util.*;
import java.io.*;
import java.net.*;

public class FileServer
{
public static void main(String a[])
{
try
{
ServerSocket s=new ServerSocket(95);
System.out.println("Server Waiting For The Client");
Socket cs=s.accept();
BufferedReader fromclient=new BufferedReader(new
InputStreamReader(cs.getInputStream()));
String file=fromclient.readLine(); System.out.println("The file requested:"+file); int len,size;
FileInputStream fin=new FileInputStream(file);
len=fin.available();
System.out.println("File Size="+len);
byte myarr[]=new byte[512];
int tot=0;
cs.getOutputStream().flush();
OutputStream output = cs.getOutptStream();
System.out.println("File transfer begins. ...");
while((size= fin.read(myarr)) != -1)

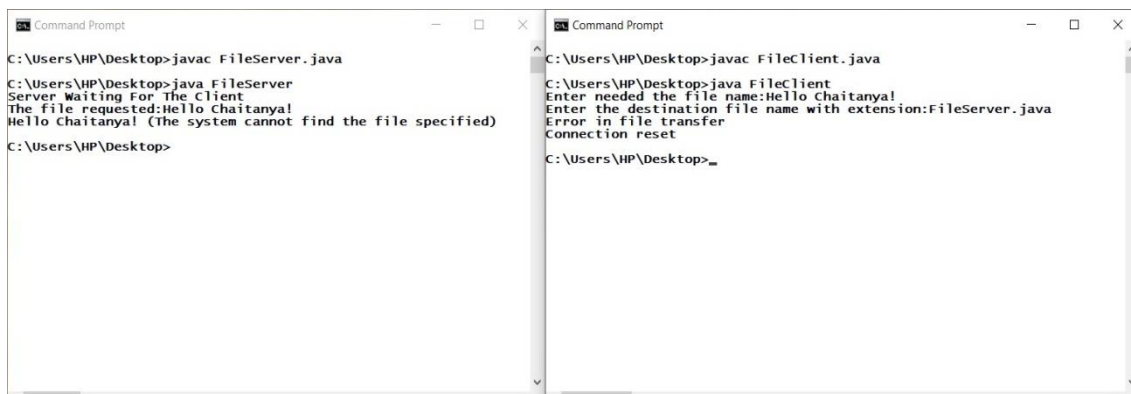
```

```

{
output.write(myarr,0,size);
tot=tot+size;
}
System.out.println("File transfer complete. ... Total Bytes transfered:"+tot);
fin.close();
fromclient.close();
output.close();
cs.close();
}
catch(IOException e)
{
System.out.println(e.getMessage());
}
}
}
}

```

Output:



```

C:\Users\HP\Desktop>javac FileServer.java
C:\Users\HP\Desktop>java FileServer
Server Waiting For The Client
The File requested:Hello Chaitanya!
Hello Chaitanya! (The system cannot find the file specified)
C:\Users\HP\Desktop>

C:\Users\HP\Desktop>javac FileClient.java
C:\Users\HP\Desktop>java FileClient
Enter needed the file name:Hello Chaitanya!
Enter the destination file name with extension:FileServer.java
Error in file transfer
Connection reset
C:\Users\HP\Desktop>

```

1.recursive algorithm(Factorial).

```
//Java Program to find the Factorial of a Number import java.util.*; public class Main
{ static int factorial(int n){
    if (n == 0) return 1;
    else
        return(n * factorial(n-1));
} public static void main(String []args)
{
    //Take input from the user
    //Create an instance of the Scanner Class
    Scanner sc=new Scanner(System.in);
    //Declare and Initialize the variable int n,fact=1;
    int n=sc.nextInt(); fact =
    factorial(number);
    System.out.println("Factorial of "+n+" is: "+fact);
}
}
```

Output:

Enter the number: 5

Factorial of the number: 120

Pigeonhole principle.

```
/* Java program to implement Pigeonhole Sort */ import
java.lang.*; import java.util.*; public class GFG {

    public static void pigeonhole_sort(int arr[], int n) { int min
    = arr[0]; int max = arr[0]; int range, i, j, index; for (int a =
    0; a < n; a++) { if (arr[a] > max)

    max = arr[a]; if (arr[a] < min)
    min = arr[a];

    }

    range = max - min + 1; int[] phole = new
    int[range]; Arrays.fill(phole, 0); for (i = 0; i
    < n; i++) phole[arr[i] - min]++; index = 0;
    for (j = 0; j < range; j++)

    while (phole[j]-- > 0) arr[index++] = j +
    min;

    } public static void main(String[] args)

    {

    GFG sort = new GFG();

    int[] arr = { 8, 3, 2, 7, 4, 6, 8 };

    System.out.print("Sorted order is : ");
    sort.pigeonhole_sort(arr, arr.length);

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

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

    }

    }
```

Output:

Sorted order is : 2 3 4 6 7 8 8

Binomial Coefficient. Program-i

```
public class BinomialDistribution { public static void main(String[] args) { int n =
Integer.parseInt(args[0]); double[][] binomial = new double[n+1][];

// initialize first row binomial[1] = new double[1 + 2]; binomial[1][1] = 1.0;

// fill in coefficients of binomial distribution for (int i = 2; i <= n; i++) { binomial[i] =
new double[i+2]; for (int k = 1; k < binomial[i].length - 1; k++) binomial[i][k] = 0.5 *
(binomial[i-1][k-1] + binomial[i-
1][k]);
}

// print binomial coefficients for (int i = 1; i <= n; i++) {
for (int k = 1; k < binomial[i].length - 1; k++) {
System.out.print(binomial[i][k] + " ");
}
System.out.println();
}
}
}
```

Program-ii

```
import java.io.*; class GFG {
// function to calculate factorial of // a number
static int factorial(int n)
{ int f = 1; for (int i = 2; i <= n; i++) f *= i;
return f;
}

// function to print the series static void series(int A, int X, int n) {
// calculating the value of n! int nFact = factorial(n); // loop to display the series for
(int i = 0; i < n + 1; i++) {
// For calculating the // value of nCr int niFact = factorial(n - i); int iFact = factorial(i);
// calculating the value of
```

```
// A to the power k and X to // the power k int aPow = (int)Math.pow(A, n - i); int  
xPow = (int)Math.pow(X, i); // display the series  
System.out.print((nFact * aPow * xPow)  
/ (niFact * iFact) + " ");  
}  
}  
  
// main function started public static void main(String[] args)  
{ int A = 3, X = 4, n = 5;  
series(A, X, n);  
}  
}
```

Output:

243 1620 4320 5760 3840 1024

.Reccurence Relations(Towers of Hanoi).

I Java program to solve tower of hanoi

```
class GFGG
{
//Java recursive function to solve tower of hanoi puzzle static void towerOfHanoi(int n, char
    from_rod, char to_rod, char aux_rod)
{ if (n== 1)
{
System.out.println("Move disk 1 from rod "+ from_rod+ " to rod" +to_rod); return;
}
towerOfHanoi(n-1, from_rod, aux_rod, to_rod);
System.out.println("Move disk " +n+ " from rod " + from_rod+ " to rod "+to_rod); towerOfHanoi(n-1,
aux_rod, to_rod, from_rod);
}

// Driver method public static void main(String
args[])
{
int n = 4;// Number of disks
towerOfHanoi(n, 'A', 'C', 'B');//A, B and C are names of rods
}
}
```

Out put:

Move disk 1 from rod A to rod B

Move disk 2 from rod A to rod C

Move disk 1 from rod B to rodC

Move disk 3 from rod A to rod B

Move disk 1 from rod C to rod A

Move disk 2 from rod C to rod B

Move disk 1 from rod A to rod B

Move disk 4 from rod A to rod C

Move disk 1 from rodB to rod C

Move disk 2 from rodB to rod A

Move disk 1 from rod C to rod A

Move disk 3 from rod B to rod C

Move disk 1 from rod A to rod B

Move disk 2 from rod A to rod C

Move disk 1 from rod B to rod C

Divide and conquer

```
import java.util.Arrays; public class
MergeSortTopDown { public int[] sort(int[] arr) {
int n = arr.length; if (n < 2) { return arr;
}

// divide the array in half int[] left =
Arrays.copyOfRange(arr, 0, n/2); int[] right =
Arrays.copyOfRange(arr, n/2, n);

// sort/conquer each half int[] sortedLeft =
sort(left); int[] sortedRight = sort(right);

// merge/combine the sorted halves return
merge(sortedLeft, sortedRight, n);
}

private int[] merge(int[] left, int[] right, int n) { int[] result =
new int[n]; int l = 0; int r = 0; for (int i = 0; i < n; i++) { if (l <
left.length && (r >= right.length || left[l] <
right[r])) {
result[i] = left[l]; l++; } else {
result[i] = right[r]; r++;
}
} return result;
} public static void main(String[] args) { int[] a = {7, 9,
2, 3, 1};
a = new MergeSortTopDown().sort(a);
Arrays.stream(a).forEach(System.out::println);
}
}
```

```

.import java.util.Scanner; 6.
7. /** Class GraphColoring */
8. public class GraphColoring
9. {
10. private int V, numOfColors;
11. private int[] color; 12. private int[][] graph; 13.
14. /** Function to assign color */
15. public void graphColor(int[][] g, int noc)
16. {
17. V = g.length;
18. numOfColors = noc;
19. color = new int[V]; 20. graph = g; 21.
22. try
23. {
24. solve(0);
25. System.out.println("No solution");
26. }
27. catch (Exception e)
28. {
29. System.out.println("\nSolution exists ");
30. display();
31. }
32. }
33. /** function to assign colors recursively */
34. public void solve(int v) throws Exception
35. {
36. /** base case - solution found */
37. if (v == V)
38. throw new Exception("Solution found");
39. /** try all colours */
40. for (int c = 1; c <= numOfColors; c++)
41. {
42. if (isPossible(v, c))
43. {
44. /** assign and proceed with next vertex */
45. color[v] = c;
46. solve(v + 1);
47. /** wrong assignement */
48. color[v] = 0;
49. } 50. }
51. }
52. /** function to check if it is valid to allot that color to vertex */
53. public boolean isPossible(int v, int c)

```



```

54. {
55. for (int i = 0; i < V; i++)
56. if (graph[v][i] == 1 && c == color[i])
57. return false;
58. return true;
59. }
60. /** display solution */
61. public void display()
62. {
63. System.out.print("\nColors : ");
64. for (int i = 0; i < V; i++)
65. System.out.print(color[i] + " ");
66. System.out.println();
67. }
68. /** Main function */
69. public static void main (String[] args)
70. {
71. Scanner scan = new Scanner(System.in);
72. System.out.println("Graph Coloring Algorithm Test\n");
73. /** Make an object of GraphColoring class */74. GraphColoring gc = new GraphColoring(); 75.
76. /** Accept number of vertices */
77. System.out.println("Enter number of vertices\n");78. int V = scan.nextInt(); 79.
80. /** get graph */
81. System.out.println("\nEnter matrix\n");
82. int[][] graph = new int[V][V];
83. for (int i = 0; i < V; i++)
84. for (int j = 0; j < V; j++) 85. graph[i][j] = scan.nextInt(); 86.
87. System.out.println("\nEnter number of colors"); 88. int c =
scan.nextInt(); 89.
90. gc.graphColor(graph, c); 91.
92. }
93. }

```

Graph Coloring Algorithm Test

Enter number of vertices

10

Enter matrix

0 1 0 0 0 1 0 0 0 0

1 0 1 0 0 0 1 0 0 0

0 1 0 1 0 0 0 1 0 0

0 0 1 0 1 0 0 0 1 0

1 0 0 1 0 0 0 0 0 1

1 0 0 0 0 0 0 1 1 0

0100000011

0010010001

0001011000

0000101100

Enter number of colors.

1.To write a java program to implement operators and conditional statements

```
Import java.util.*;

Class Experiment{

Public static void main(String args[]){

Int a,b,result=0,choice=0;

Scanner sc = new Scanner(System.in);

System.out.println("Enter any two numbers :");

a = sc.nextInt();

b = sc.nextInt();

if(a==0&&b==0){

System.out.println("The Result will be :"+result);}

Else{

System.out.println("Please select your option :");

System.out.println("1.addition\n 2.substraction\n 3.multiplication\n 4.division\n");

Choice sc.nextInt();

Switch(choice){

Case 1: result =a+b; break;

Case 2: result =a-b;break;

Case 3: result =a*b;break;

Case 4: result =a/b;break;

Default:System.out.println("Invalid option");}}

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

}}
```

Output:

Enter any two numbers

10

2

Please select your option

1.addition

2.substraction

3.multipliction

4.division

3

The result is : 20