

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

{

```
    for (j=0 ; j<2 , j++) {
```

```
        mat2[i][j] = sc.nextInt();
```

}

}

```
System.out.println("Adding matrix 1 and 2 elements :");
```

```
for (i=0 , i<2 , i++) {
```

```
    for (j=0 , j<2 , j++) {
```

```
        mat3[i][j] = mat1[i][j] + mat2[i][j];
```

}

}

```
System.out.println("Addition done ");
```

```
System.out.println("New matrix is :");
```

```
for (i=0 ; i<2 , i++) {
```

```
    for (j=0 , j<2 , j++) {
```

```
        System.out.println(" mat3[" + i + "][ " + j + " ] " );
```

}

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

}

y

y

CONCLUSION: The code has executed successfully.

C) AIM: To write a java program for multiplying two matrices and print the product for the same.

THEORY: The total number of elements that can be stored in multi-dimensional array can be calculated by multiplying the size of all the dimensions.

IMPLEMENTATION OF CODE:

```
import java.util.Scanner;
public class prac6c {
    public void static void main (String args[]) {
        int i, j, k;
        Scanner sc = new Scanner (System.in);
        int n= sc.nextInt();
        int mat1= new int [n][n];
        int mat2= new int [n][n];
        int mat3= new int [n][n];
        System.out.println ("enter matrix 1 elements ");
        for (i=0 , i<n ; i++) {
            for (j=0 ; j<n ; j++) {
                mat1[n][j]= sc.nextInt();
            }
        }
    }
}
```

```
System.out.println ("enter matrix 2 elements ");
for (i=0 , i<n , i++) {
    for (j=0 , j<n , j++)
```

PRACTICAL NO: 7 Vectors and Multithreading.

a) AIM: To write a java program to implement the vectors.

THEORY: Vector is like dynamic array which can grow or shrink in size. Unlike array we can store n-number of elements in it as there is no size limit. It is found in `java.util` package & implements `List` interface.

IMPLEMENTATION OF CODE:

```

import java.util.Scanner;
public class VectorDemo {
    public static void main (String args []) {
        Vector v = new Vector (10, 2);
        Integer i = new Integer (100);
        Double d = new Double (45.699);
        Long l = new Long (5042569);
        String s = new String ("Hello");
        v.addElement (i);
        v.addElement (d);
        v.addElement (l);
        v.addElement (s);
        System.out.println ("Capacity of Vector " + v.capacity ());
        int size = v.size ();
        System.out.println ("Size of vector " + size);
    }
}

```

```

System.out.println("Displaying Vector Elements");
for (int j=0 ; j < size ; j++)
{
    System.out.println(" " + v.elementAt(j));
}
v.removeElementAt(2);
System.out.println("Removing Element at second index");
v.removeElement("Hello");
System.out.println("Removing element Hello");
System.out.println("First element in vector = " + v.firstElement());
System.out.println("Last element in vector = " + v.lastElement());
// v.removeAllElements();
// System.out.println("Removing all elements");
boolean b = v.isEmpty();
System.out.println(" Is vector empty " + b);

```

CONCLUSION: The code has been executed successfully.

AIM: To write java program to implement multithreading.

THEORY: Multithreading is a process of executing multiple threads simultaneously. A thread is a lightweight subprocess, the smallest unit of processing. Multiprocessing and multithreading are used to achieve multitasking.

IMPLEMENTATION OF CODE:

```
class MTDemo2 implements Runnable {
```

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

```
System.out.println ("My thread is in running state.");
```

3

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

```
MTDemo2 obj = new MTDemo2();
```

```
Thread tobj = new Thread (obj);
```

```
tobj.start();
```

4

CONCLUSION: The multithreading demo has been executed successfully.

PRACTICAL NO: 8

A) AIM : To write a java program to open a file and display the contents in the console window.

THEORY: In java, file is an abstract data type & named location used to store related information is known as File. There are several file operations like creating new file, getting information about file, writing into a file.

IMPLEMENTATION OF CODE:

```

import java.io.*;
public class Prac8a
{
    public static void main (String args[])
        throws
            FileNotFoundException, IOException
    {
        InputStream is = new FileInputStream ("c:\\\\Users\\\\Lenovo\\\\Desktop\\\\prac8.txt");
        int count = is.available();
        byte a [] = new byte [count];
        is.read (a);
        for (byte b : a)
        {
            char k = (char)b;
            System.out.print (b); yy
        }
    }
}

```

CONCLUSION: The code has been executed successfully.

6) AIM: To write a java program to copy the contents from one file to other file.

THEORY: A series of data is referred to as stream.

In java, there are two types Byte stream and character stream. Byte stream is mainly involved with byte data. A file handling process in which input is provided and executed with byte data.

IMPLEMENTATION OF CODE!

```

import java.io.*;
public class Prac86
{
    public class static void main (String args[]) throws
        FileNotFoundException, IOException
    {
        FileInputStream fis = null;
        FileOutputStream fos = null;
        try
        {
            File infile = new File ("C:\\Users\\lenovo\\Desktop\\1.txt");
            File outfile = new File ("C:\\Users\\lenovo\\Desktop\\2.txt");
            fis = new FileInputStream (infile);
            fos = new FileOutputStream (outfile);
            byte buff [] = new byte [1024];
            int length;
            while ((length = fis.read (buff)) > 0)
            {
                fos.write (buff, 0, length);
            }
        }
        catch (Exception e)
        {
            e.printStackTrace ();
        }
    }
}

```

```
fos.write (buff, 0, length);  
y  
fis.close();  
fos.close();  
System.out.println ("CONTENT COPIED");  
y  
catch (Exception e)  
{  
    System.out.println ("error");  
    y  
    y  
    y}
```

CONCLUSION: The code has been executed successfully.

AIM: To write a java program to read the student data from user and store it in the file.

THEORY: Java FileOutputStream is an output stream used for writing data to file. If you have to write primitive values into a file, use FileOutputStream class. FileInputStream are byte stream that read and write data in binary format.

IMPLEMENTATION OF CODE:

```

import java.io.*;
import java.util.Scanner;
public class prac8c {
    public static void main (String [] args) throws IOException
        FileNotFoundException, IOException
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("enter name");
        String s1 = sc.nextLine ();
        System.out.println ("phone number and address");
        String s2 = sc.nextLine ();
        FileOutputStream fos = new FileOutputStream
            ("C:\\Users\\lenovo\\Desktop\\2.txt");
        byte b1[] = s1.getBytes ();
        fos.write (b1);
    }
}

```

```
byte b2[] = s2.getBytes();  
fos.write(b2);
```

```
fos.close();
```

```
System.out.println("file created and data store");
```

```
}
```

```
}
```

CONCLUSION: The code has been executed successfully.

PRACTICAL NO: ⑨

A AIM: Design a AWT program to print the factorial for an input value.

THEORY: AWT stands for Abstract Window Toolkit is an Application programming interface (API) for creating graphical user interface (GUI) in Java. It allows Java programmers to develop window-based application.

IMPLEMENTATION OF CODE:

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

```
public class cjava extends Frame implements
ActionListener
```

d

Label l1, l2;

TextField t1, t2;

Button b1, b2;

cjava()

d

t1 = new TextField(8);

t2 = new TextField(10);

b1 = new Button("Factorial");

b2 = new Button("clear");

l1 = new Label("Enter number");

```

l2 = new Label ("Result");
setLayout (new GridLayout( 3,2));
add(l1);
add(t1);
add(l2);
add(t2);
add(b1);
add(b2);
setSize (200 , 200);
setVisible (true);
b1.add ActionListener (this);
b2.add ActionListener (this);
addWindowListener (new WindowAdapter()

```

{

```
public void windowClosing (WindowEvent we)
```

{

```
System.exit (0);
```

}

}

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

{

```
new Cjgal();
```

}

```
public void actionPerformed (ActionEvent ae)
```

{

```
if (ae.getSource () == b1)
```

```
int num , fact , i;
```

```
String result;  
fact = 1;  
num = Integer.parseInt(t1.getText());  
for (i=1; i <= num; i++)  
    d  
    fact = fact * i;  
    y  
result = String.valueOf(fact);  
t2.setText(result);  
y  
else  
d  
t1.setText("");  
t2.setText("");  
y  
y  
y
```

CONCLUSION: GUI using AWT has been created successfully.

AIM: To design an AWT program to perform various string operations like reverse string, string concatenation etc.

THEORY: AWT provides various components like button, label, checkbox etc. used as objects in java program. AWT components use the resources of operating system i.e. they are platform dependent.

IMPLEMENTATION OF CODE:

```
import java.awt.*;
import java.awt.event.*;
public class Prac96 extends Frame implements ActionListener
```

Label l1, l2, l3,

TextField t1, t2, t3,

Button b1, b2, b3, b4;

Prac96() {

l1 = new Label ("1st String");

l2 = new Label ("2nd string");

l3 = new Label ("result");

t1 = new TextField (10);

t2 = new TextField (10);

t3 = new TextField (10);

b1 = new Button ("reverse for 1st");

b2 = new Button ("concatenate for 1st");

b3 = new Button ("length of 1st");

```

b4 = new JButton("CLEAR"),
setLayout(new GridLayout(5,2)),
add(l1);
add(t1);
add(t2);
add(l3);
add(t3);
add(b1);
add(b2);
add(b3);
add(b4);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
setVisible(true);
addWindowListener(new WindowAdapter()
{
    public void windowClosing(WindowEvent we)
    {
        System.exit(0);
    }
});
public static void main(String args[])
{
    new Prac9b();
}

```

public void actionPerformed(ActionEvent ae)

{

String result;

StringBuilder str,

if (ae.getSource() == b1)

{

str = new StringBuilder();

str.append(t1.getText());

t3.setText(str.toString());

str.reverse();

y

else if (ae.getSource() == b2),

{

result = t1.getText() + t2.getText();

t3.setText(result);

y

else if (ae.getSource() == b3)

{

str = new StringBuilder();

str.append(t1.getText());

int a = str.length();

t3.setText(String.valueOf(a));

y

else

{

t1.setText(" ");

t2.setText(" ");

t3.setText(" ");

y y

4

CONCLUSION: The code has executed successfully - executed.

PRACTICAL NO: 10

AIM: To write Java program to implement exception handling.

THEORY: Exception Handling in Java is one of the powerful mechanism to handle the runtime errors so that the normal flow of the application can be maintained.

IMPLEMENTATION OF CODE:

```

class practice {
    public static void main (String args [])
    {
        System.out.println ("program starts");
        try
        {
            int num = 32 / 0;
            // int num = 0 / 0;           // arithmetic
            // float num = 32.35 / 0;   // prints infinity
            // float num = 0.0 / 0;     // prints NAN
            System.out.println ("num = " + num);
        }
        catch (ArithmeticException e)
        {
            System.out.println ("division by zero");
            System.out.println (e.getMessage()); // by zero
        }
    }
}

```

y
System.out.println("Program Ends");
y

CONCLUSION : The code has been executed
successfully.

INDEX

EXPERIMENTS

Sr. No.	Title	Page No.	Date	Remark	Teacher's Signature
	PRACTICAL NO: 1				
A)	Write a java program that takes numbers as input & prints its multiplication table.				
B)	Write java program to display	*	*	*	*
		*	*	*	*
		*	*	*	*
		*	*	*	*
C)	Write java program to find area & perimeter of a circle.				
	PRACTICAL NO: 2				
A)	Write java program to add two binary numbers				
B)	Write java program to convert a decimal number to binary & vice versa.				
C)	Write java program to reverse a string				
	PRACTICAL NO: 3				

INDEX

EXPERIMENTS

Sr. No.	Title	Page No.	Date	Remark	Teacher's Signature
a)	Write java program to count letters, spaces numbers & other characters of input string.				
b)	Implement a java function that calculates the sum of digits for given char array consisting of digits '0' to 9. The function should return the digit sum as long value.				
c)	Find smallest & largest element in an array.				
PRACTICAL NO : 4					
A)	Design a class SortData that contains method asec() & desc().				
B)	Design a class that demonstrates use of constructor and destructor				
C)	Write java program to demonstrate implementation of abstract class				

INDEX

EXPERIMENTS

Sr. No.	Title	Page No.	Date	Remark	Teacher's Signature
	PRACTICAL NO: 5				
A)	Write a java program to implement single level inheritance				
B)	Write a java program to implement method overriding				
C)	Write java program to implement multiple inheritance				
	PRACTICAL NO: 6				
A)	Create a package . Add the necessary classes & import package in java class.				
B)	Write a java program to add matrices and find the resultant matrix.				
C)	Write java program for multiplying two matrices & printing resultant matrix.				

INDEX

EXPERIMENTS

Sr. No.	Title	Page No.	Date	Remark	Teacher's Signature
	PRACTICAL NO: 7				
A)	Write java program to implement the vectors.				
B)	Write java program to implement thread life cycle.				
C)	Write java program to implement multithreading.				
	PRACTICAL NO : 8				
A)	Write java program to open a file and display the contents in console window.				
B)	Write a java program to copy the contents of one file to other file.				
C)	Write a java program to read the student data from user & store in the file				

INDEX

EXPERIMENTS