**EXPERIMENT NO: 01**

**Date: 21/11/2021**

# AIM :

*To write a java program to check whether the input string is palindrome or*

*not.*

# ALGORITHM :

1. Start
2. Declare string variables str and rev, then initialize rev = “ ”
3. Read the string from the user and store it to the variable str
4. Calculate the length of the string and store it in the variable len
5. For the value of i = len -1 to i >=0 do
   1. rev = rev + str.charAt(i)
6. If str and rev are equal
   1. Print the entered string is palindrome
   2. Else print the entered string is not palindrome
7. Stop

# PROGRAM CODE:

import java.util.Scanner;

public class palindrome{

public static void main(String[] args)

{

String str, rev = "";

System.out.print("--------------\n\n\n");

System.out.print("Enter a string : ");

Scanner sc = new Scanner(System.in);

str = sc.nextLine();

int len = str.length();

for (int i = len - 1; i >= 0; i--)

{

rev = rev + str.charAt(i);

}

if (str.equalsIgnoreCase(rev))

{

System.out.println(str + " is a palindrome string");

}

else

{

System.out.println(str + " is not a palindrome string");

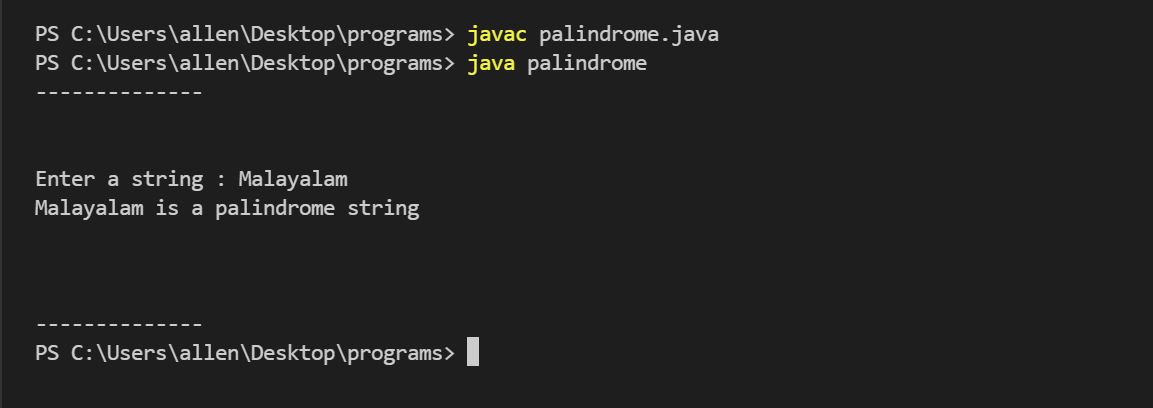
}

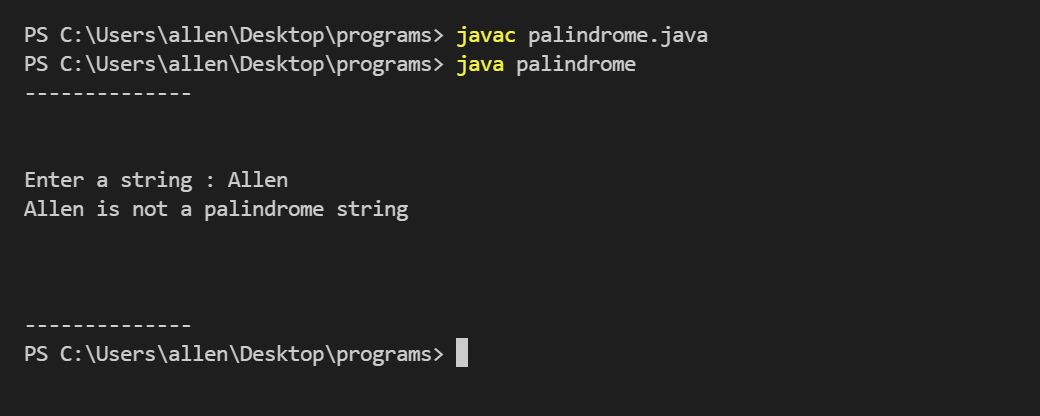
System.out.print("\n\n\n--------------");

}

}

**OUTPUT :**

****

****

**RESULT :**

The program compiled and executed successfully.

**EXPERIMENT NO: 02**

**Date: 29/11/2021**

# AIM :

*To write a java program to find the frequency of a characters in a given string*

# ALGORITHM :

1. Start
2. Declare string variables str and str1
3. Declare integer variables i, len, count and initialize count as 0
4. Declare character variables ch and ch1
5. Read the string from the user and store it to str
6. Convert the string in str to lowercase and store in str1
7. Count the length of the string and store the value in len
8. Read the character to find the frequency from the user and store it to ch
9. Convert the character in ch to lowercase and store it in ch1 10.For the value of i = 0 to i < len do

a. If(ch1 == str1.charAt(i))

* 1. Increase the value of count by 1. 11.Print the frequency of the character.

1. Stop

# PROGRAM CODE:

# import java.util.Scanner;

# public class frequency

# {

# public static void main(String[] args)

# {

# String str, str1 ;

# int i, len, count=0 ;

# char ch, ch1 ;

# System.out.print("--------------\n\n\n");

# System.out.print("Enter a string : ");

# Scanner sc = new Scanner(System.in);

# str = sc.nextLine();

# str1 = str.toLowerCase();

# len = str.length();

# System.out.print("Enter a character to find frequency : ");

# Scanner c = new Scanner(System.in);

# ch = c.next().charAt(0);

# ch1 = Character.toLowerCase(ch);

# for(i=0;i<len;i++)

# {

# if(ch1 == str1.charAt(i))

# {

# count ++;

# }

# }

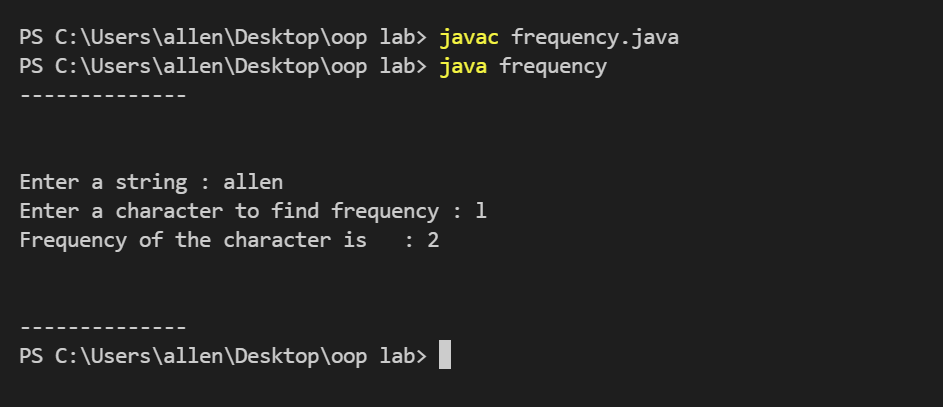
# System.out.print("Frequency of the character is : "+count);

# System.out.print("\n\n\n--------------");

# }

# }

**OUTPUT :**

****

**RESULT :**

The program compiled and executed successfully.

**EXPERIMENT NO: 03**

**Date: 29/11/2021**

# AIM :

*To write a java program to multiply two matrices.*

# ALGORITHM :

* 1. Start
  2. Declare variables i, j, k and array c[10][10]
  3. Accept the number of rows and columns for the first matrix from the user as m1 and m2
  4. An array a with size as m1 and m2 are declared
  5. Accept the number of rows and columns for the second matrix from the user as n1 and n2
  6. An array b with size as n1 and n2 are declared
  7. If m1 and n2 are equal do
     1. Read the elements of first matrix from the user
     2. Read the elements of second matrix from the user
     3. For the value of i<m1 do
        1. For the value of j<n2 do a. c[i][j] = 0

b. For the value of k<n1 do

i. c[i][j] = c[i][j] + a[i][j] \* b[i][j]

* + 1. Print the OUPUT matrix c[i][j]
  1. If m1 not equal to n2
     1. Print multiplication not possible
  2. Stop

# PROGRAM CODE:

import java.util.Scanner;

public class Matrix\_multiplication

{

public static void main(String[] args)

{

int m1,n1,m2,n2,i,j,k;

int [][]c = new int[50][50];

System.out.print("--------------\n\n\n");

System.out.println("Enter order of 1st matrix : ");

Scanner e=new Scanner(System.in);

Scanner f=new Scanner(System.in);

m1 = e.nextInt();

m2 = f.nextInt();

int a[][] = new int[m1][m1];

System.out.println("Enter order of 2nd matrix : ");

Scanner g=new Scanner(System.in);

Scanner h=new Scanner(System.in);

n1 = g.nextInt();

n2 = h.nextInt();

int b[][] = new int[n1][n1];

if(m1 != n2)

{

System.out.print("Multiplication not possible");

}

if(m1 == n2)

{

System.out.println("Enter the elements of first matrix : ");

for(i=0;i<m1;i++)

{

for(j=0;j<m2;j++)

{

Scanner s = new Scanner(System.in);

a[i][j] = s.nextInt();

}

}

System.out.println("Enter the elements of second matrix : ");

for(i=0;i<n1;i++)

{

for(j=0;j<n2;j++)

{

Scanner p = new Scanner(System.in);

b[i][j] = p.nextInt();

}

}

for(i=0;i<m1;i++)

{

for(j=0;j<n2;j++)

{

c[i][j] = 0;

for(k=0;k<n1;k++)

{

c[i][j] = c[i][j]+a[i][k]\*b[k][j];

}

}

}

System.out.println("\n Product is : ");

for(i=0;i<m1;i++)

{

System.out.print("\n");

for(j=0;j<m2;j++)

{

System.out.print(c[i][j]+"\t");

}

}

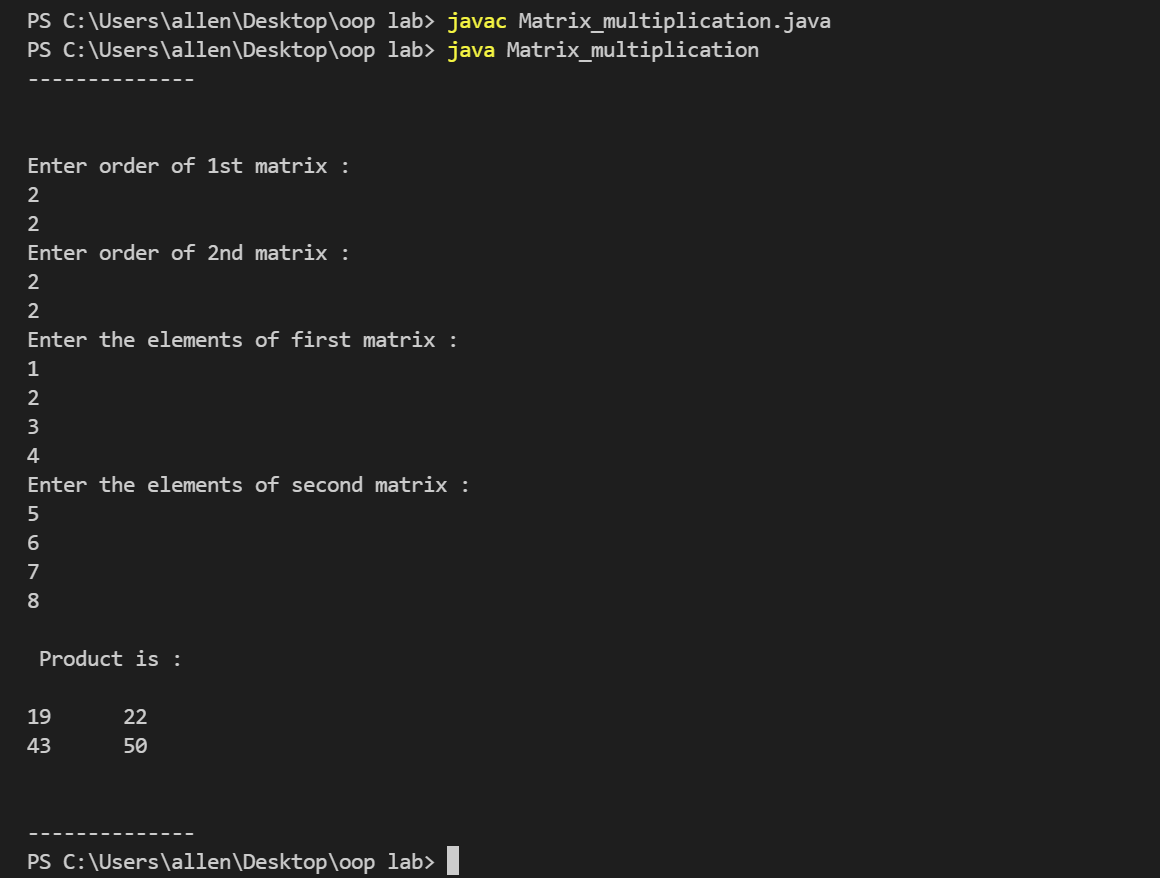
}

System.out.print("\n\n\n--------------");

}

}

# OUTPUT :



**RESULT :**

The program compiled and executed successfully.

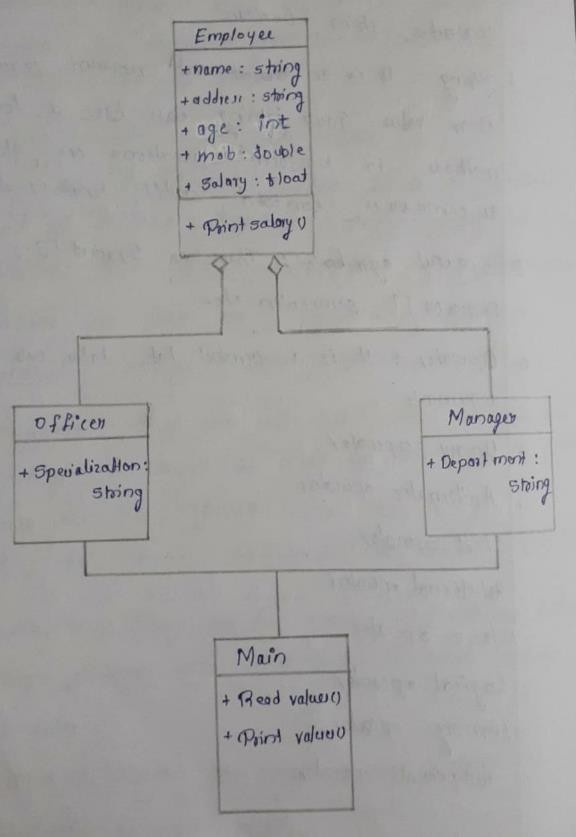
**EXPERIMENT NO: 04**

**Date: 15/01/2022**

# AIM :

To write a java program to creates a class named Employee having the following members : Name, Age, Phone number, Address, Salary. It also has a method named printSalary() which prints the salary of the Employee. Two classes Officer and Manager inherits the Employee class. The Officer and Manager classes have data members specialization and department respectively. Now, assign name, age, phone number, address and salary to an officer and a manager by making an object of both of these classes and print the same

# Class Diagram :



**PROGRAM CODE:**

import java.util.Scanner;

class Employee

{

public String name;

public String address;

public int age;

public String mob;

public float salary;

public void printsalary()

{

System.out.print("\nSalary of the employee is "+salary);

}

}

class Officer extends Employee

{

public String specialization;

}

class Manager extends Employee

{

public String department;

}

class Empinheritance{

public static void main(String args[])

{

Officer of = new Officer();

Manager mn = new Manager();

Scanner in = new Scanner(System.in);

Scanner sc = new Scanner(System.in);

System.out.print("--------------\n\n\n");

System.out.print("\n\n\tEnter the details of officer\n");

System.out.print("Enter the name : ");

of.name = sc.nextLine();

System.out.print("Enter the address : ");

of.address = sc.nextLine();

System.out.print("Enter mobile number : ");

of.mob = sc.nextLine();

System.out.print("Enter age : " );

of.age = in.nextInt();

System.out.print("Enter specialization : ");

of.specialization = sc.nextLine();

System.out.print("Enter salary : ");

of.salary = in.nextFloat();

System.out.print("\n\n\tEnter the details of manager\n");

System.out.print("Enter the name : ");

mn.name = sc.nextLine();

System.out.print("Enter the address : ");

mn.address = sc.nextLine();

System.out.print("Enter mobile number : ");

mn.mob = sc.nextLine();

System.out.print("Enter age : " );

mn.age = in.nextInt();

System.out.print("Enter department : ");

mn.department = sc.nextLine();

System.out.print("Enter salary : ");

mn.salary = in.nextFloat();

System.out.print("\n\t\t--- Printing the details ---");

System.out.print("\n\n\t The details of the officer \n\n");

System.out.print("Name : "+of.name);

System.out.print("\nAddress : "+of.address);

System.out.print("\nMobile number : "+of.mob);

System.out.print("\nspecialization : "+of.specialization);

of.printsalary();

System.out.print("\n\n\t The details of the manager \n\n");

System.out.print("Name : "+mn.name);

System.out.print("\nAddress : "+mn.address);

System.out.print("\nMobile number : "+mn.mob);

System.out.print("\nDepartment : "+mn.department);

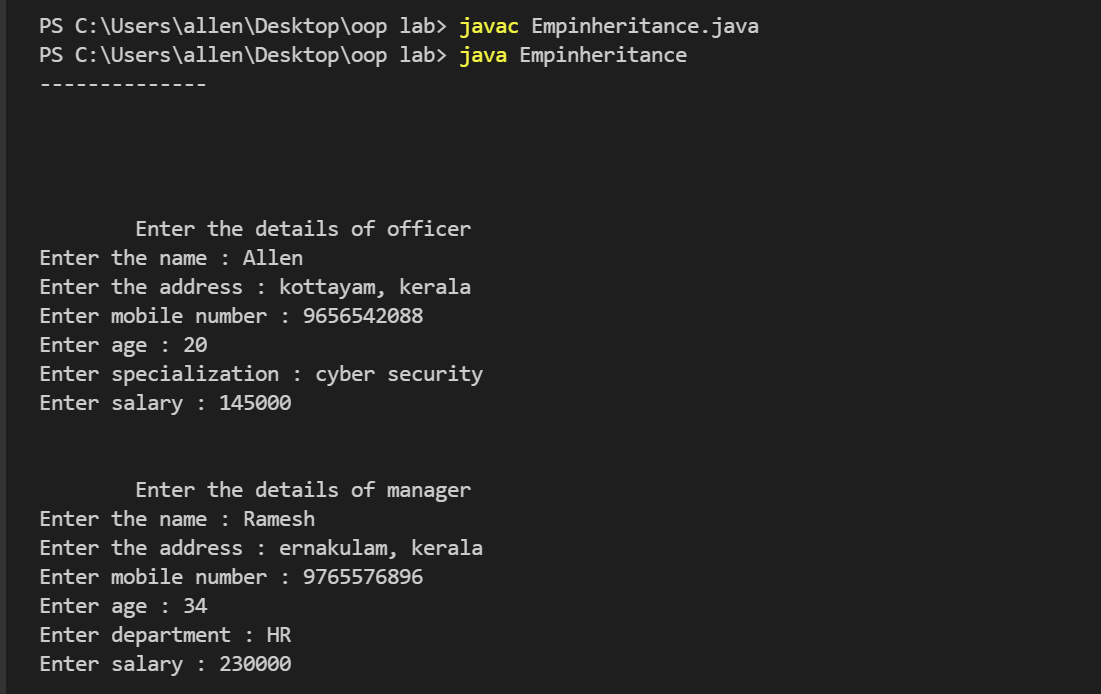
mn.printsalary();

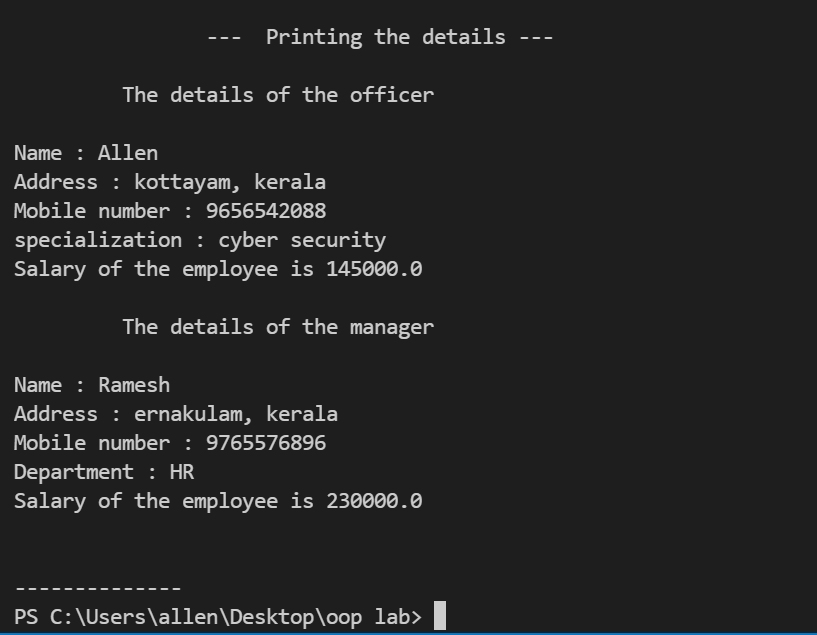
System.out.print("\n\n\n--------------");

}

}

# OUTPUT :





**RESULT :**

The program compiled and executed successfully.

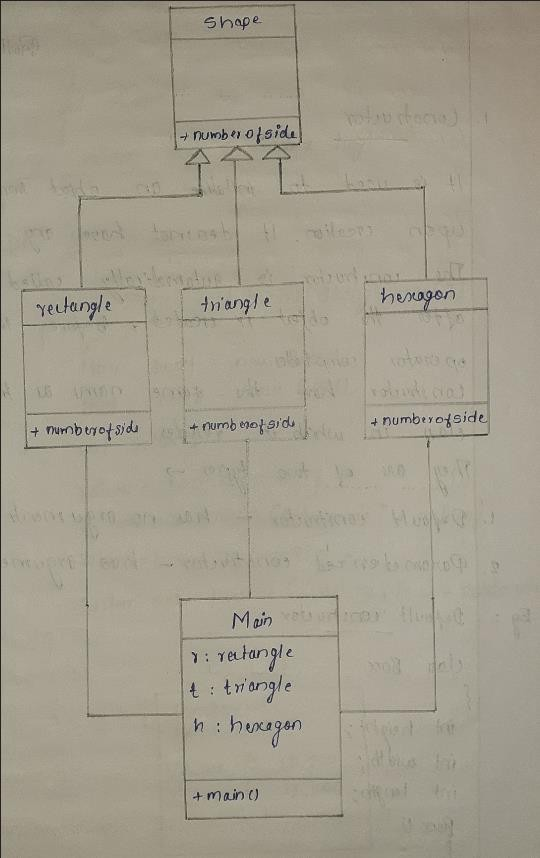
**EXPERIMENT NO: 05**

**Date: 18/01/2022**

# AIM :

*To write a java program to create an abstract class named Shape that contains an empty method named numberOfSides(). Provide three classes named Rectangle, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method numberOfSides() that shows the number of sides in the given geometrical structures.*

# Class Diagram :



**PROGRAM CODE:**

abstract class shape

{

abstract void numberOfSides();

}

class rectangle extends shape

{

public void numberOfSides()

{

System.out.println("Number of sides of rectangle : 4");

}

}

class triangle extends shape

{

public void numberOfSides()

{

System.out.println("Number of sides of triangle : 3");

}

}

class hexagon extends shape

{

public void numberOfSides()

{

System.out.println("Number of sides of hexagon : 6");

}

}

class abstractprogram

{

public static void main(String args[])

{

rectangle r=new rectangle();

triangle t=new triangle();

hexagon h=new hexagon();

System.out.print("--------------\n\n\n");

r.numberOfSides();

t.numberOfSides();

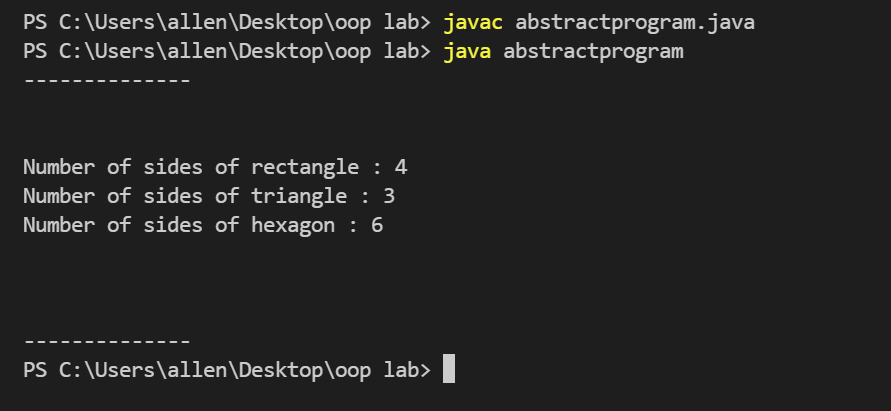
h.numberOfSides();

System.out.print("\n\n\n--------------");

}

}

**OUTPUT :**

****

**RESULT :**

The program compiled and executed successfully.

**EXPERIMENT NO: 06**

**Date: 18/01/2022**

# AIM :

*To write a java program to demonstrate the use of garbage collector.*

# ALGORITHM :

1. Start
2. Class garbagetest is created
   1. Create method finalize()
      1. Print Garbage collected
3. Object t1 and t2 are created
4. T1 = null
5. T2 = null
6. Sytem.gc() is invoked
7. Stop

# PROGRAM CODE:

public class garbagetest

{

public void finalize()

{

System.out.println("Garbage collected");

}

public static void main(String args[])

{

garbagetest t1 = new garbagetest();

garbagetest t2 = new garbagetest();

t1 = null;

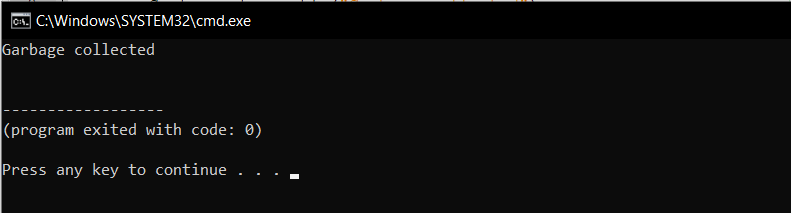
t2 = null;

System.gc();

}

}

# OUTPUT :



**RESULT :**

The program compiled and executed successfully.

**EXPERIMENT NO: 07**

**Date: 31/01/2022**

# AIM :

*To write a java program to implement file handling with reader/writer.*

# ALGORITHM :

1. Start
2. Main function is invoked
3. In the try function do the following
   1. Accept the text from the user as str
   2. Create an object file for filewiter
   3. Write the string to the file
   4. Close the file
4. If an exception occurred catch the exception
5. In another try function do the following
   1. Declare a character array ch
   2. An object rd is created for filereader
   3. Print the contents in the file
   4. Close the file
6. If an exception occurred catch the exception
7. Stop

# PROGRAM CODE:

import java.util.Scanner;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.FileNotFoundException;

class filewrre

{

public static void main(String args[])

{

try

{

System.out.print("--------------\n\n\n");

System.out.print("Enter the text to insert : ");

Scanner s = new Scanner(System.in);

String str = s.nextLine();

FileWriter file = new FileWriter("sample.txt");

file.write(str);

file.close();

System.out.print("\nText written succesfully");

}

catch(Exception e)

{

System.out.print("\nError occurred");

e.printStackTrace();

}

try

{

char ch[] = new char[100];

FileReader rd = new FileReader("sample.txt");

rd.read(ch);

System.out.println("\n\nThe contents in the file is \n");

System.out.println(ch);

System.out.print("\n\n\n--------------");

rd.close();

}

catch(Exception e)

{

System.out.print("\nError occurred");

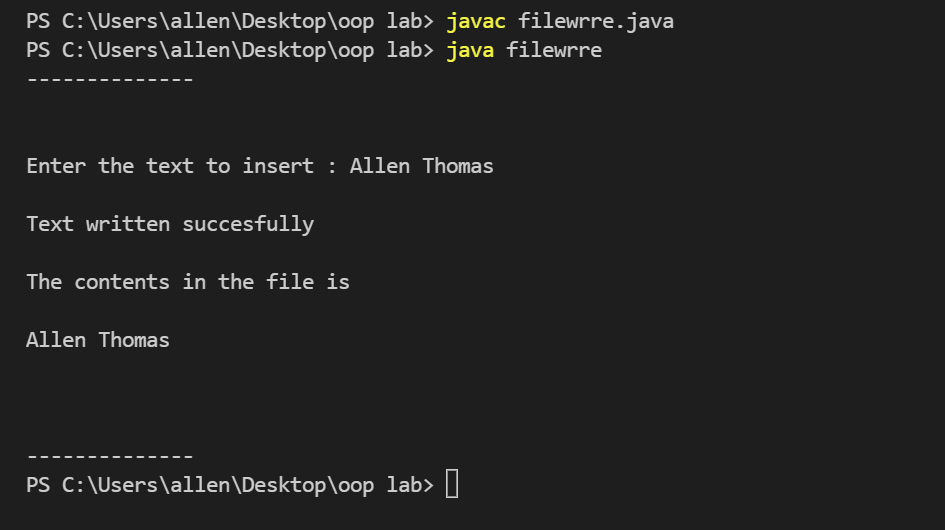
e.printStackTrace();

}

}

}

# OUTPUT :



**RESULT :**

The program compiled and executed successfully.

**EXPERIMENT NO: 08**

**Date: 31/01/2022**

# AIM :

*To write a java program that read from a file and write to file by handling all file related exceptions.*

# ALGORITHM :

1. Start
2. Main function is invoked
3. In the try function do the following
   1. Accept file name from the user
   2. Create a new file with the file name
   3. If successfully created
      1. Print file created with file name
   4. Else
      1. Print file already exists
4. If an exception occurred catch the exception
5. In another try function do the following
   1. Accept the text from the user as str
   2. Create an object file for filewiter
   3. Write the string to the file
   4. Close the file
6. If an exception occurred catch the exception
7. In another try function do the following
   1. Declare a character array ch
   2. An object rd is created for filereader
   3. Print the contents in the file
   4. Close the file
8. If an exception occurred catch the exception
9. Stop

# PROGRAM CODE:

import java.util.Scanner;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.FileNotFoundException;

class filecrwrre

{

public static void main(String args[])

{

Scanner s = new Scanner(System.in);

try

{

System.out.print("--------------\n\n\n");

System.out.print("Enter file Name : ");

String filename = s.nextLine();

File create = new File(filename);

if(create.createNewFile())

{

System.out.println("File created succesfully "+create.getName());

}

else

{

System.out.println("File already Exist");

}

}

catch (Exception e)

{

System.out.println("An error occurred.");

e.printStackTrace();

}

try

{

System.out.print("\nEnter the text to insert : ");

String str = s.nextLine();

FileWriter file = new FileWriter("filename");

file.write(str);

file.close();

System.out.print("\nText written succesfully");

}

catch(Exception e)

{

System.out.print("\nError occurred");

e.printStackTrace();

}

try

{

char ch[] = new char[100];

FileReader rd = new FileReader("filename");

rd.read(ch);

System.out.println("\n\nThe contents in the file is \n");

System.out.println(ch);

System.out.print("\n\n\n--------------");

rd.close();

}

catch(Exception e)

{

System.out.print("\nError occurred");

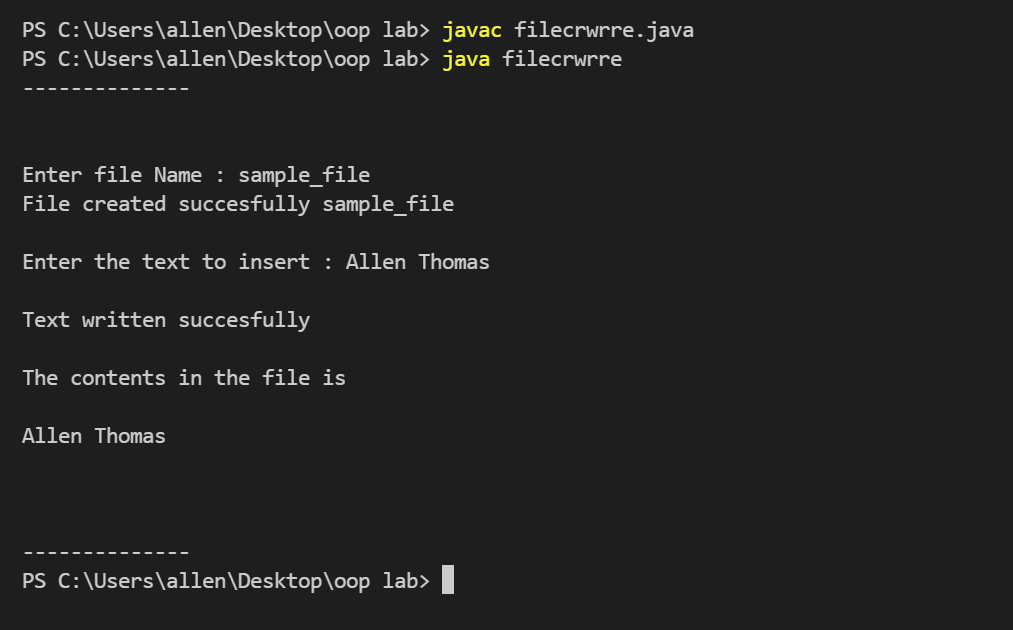
e.printStackTrace();

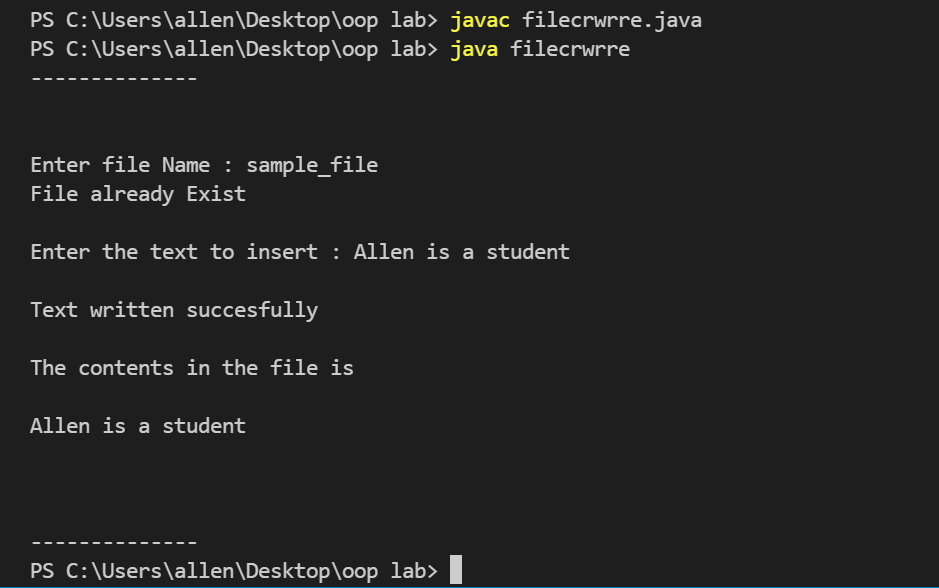
}

}

}

# OUTPUT :

****

****

**RESULT :**

The program compiled and executed successfully.

**EXPERIMENT NO: 09**

**Date: 07/02/2022**

# AIM :

*To write a java program that reads a line of integers, and then displays each integer, and the sum of all the integers*

# ALGORITHM :

1. Start
2. Main function is invoked
3. Initialize integer variables num and sum
4. Accept the maximum limit from the user
5. Accept the string numbers separated with space
6. Create a string tokenizer object str
7. While (str.hasMoreTokens())
   1. Sum = Sum + Integer.parseInt(str.nextToken()
8. Print the sum
9. Stop

# PROGRAM CODE:

import java.util.Scanner;

import java.util.StringTokenizer;

class tokenizer

{

public static void main(String args[])

{

int num, sum = 0;

Scanner sc = new Scanner(System.in);

Scanner pd = new Scanner(System.in);

System.out.print("--------------\n\n\n");

System.out.print("Enter the limit : ");

num = sc.nextInt();

System.out.print("\nEnter "+num+" numbers seperated by space : ");

String list = pd.nextLine();

StringTokenizer str = new StringTokenizer(list," ");

while(str.hasMoreTokens())

{

sum = sum + Integer.parseInt(str.nextToken());

}

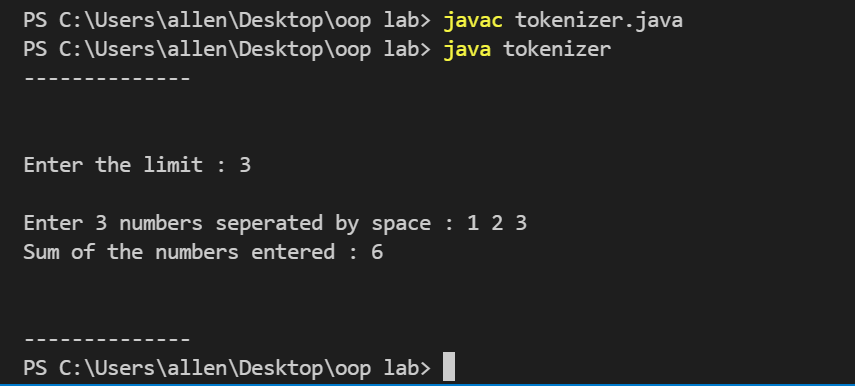
System.out.print("Sum of the numbers entered : "+sum);

System.out.print("\n\n\n--------------");

}

}

**OUPUT :**

****

**RESULT :**

The program compiled and executed successfully.

**EXPERIMENT NO: 10**

**Date: 07/02/2022**

# AIM :

*To write a java program that shows the usage of try, catch, throws and finally*

# ALGORITHM :

1. Start
2. Try block is invokked
   1. Input two numbers a and b
   2. Ans = a/b
   3. Print ans
3. If an exception occurs catch the exception
   1. Print e.getmessage()
4. Finally
   1. Print operation ended
5. Stop

# PROGRAM CODE:

import java.util.Scanner;

public class trycatch

{

public static void main(String args[])

{

try

{

System.out.print("--------------\n\n\n");

System.out.print("Enter two numbers to divide : ");

Scanner s=new Scanner(System.in);

int a = s.nextInt();

int b = s.nextInt();

float ans = (float)a/b;

System.out.println("Result is : "+ans);

}

catch(ArithmeticException e)

{

System.out.print(e.getMessage());

}

finally

{

System.out.println("Operation Ended");

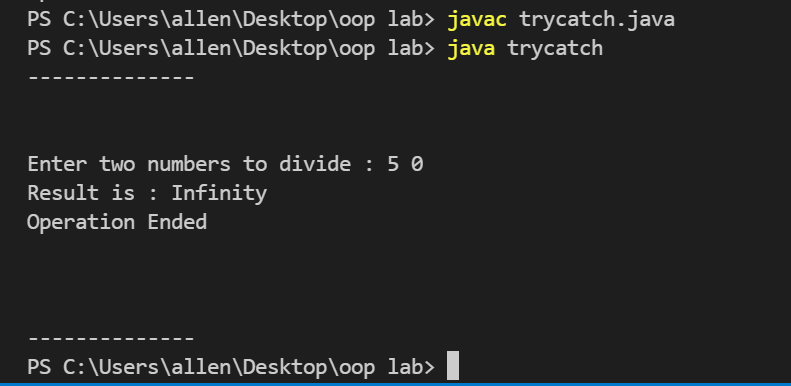
System.out.print("\n\n\n--------------");

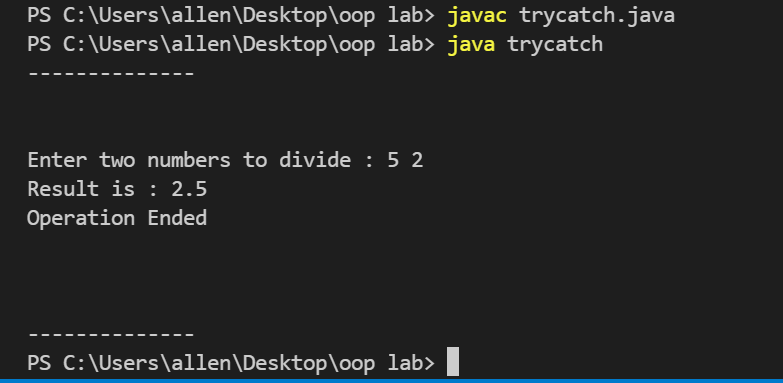
}

}

}

**OUTPUT :**

****

****

**RESULT :**

The program compiled and executed successfully.

**EXPERIMENT NO: 11**

**Date: 14/02/2022**

# AIM :

*To write a java program that implements a multi-threaded program which has three threads. First thread generates a random integer every 1 second. If the value is even, second thread computes the square of the number and prints. If the value is odd the third thread will print the value of cube of the number.*

# ALGORITHM :

1. Start
2. Main function is invoked
   1. Randomizer rz = new Randomizer
   2. Start method is called
3. Define thread class randomizer
   1. Run method is defined
      1. In this run method a random number id generated
      2. For i = 0 to i <10 do
         1. N = r.nextInt(100) 2. If n % 2 == 0
            1. An object for even class is created and start method is called
4. Else
   1. An object of odd class is created and start method is called
5. Define thread class even
   1. Run method is defined
      1. Display the square of the number
6. Define thread class odd
   1. Run method is defined
      1. Display the cube of the number
7. Stop

# PROGRAM CODE:

import java.util.Random;

class Randomizer extends Thread

{

public void run()

{

Random r = new Random();

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

{

int n = r.nextInt(100);

if (n%2==0)

{

new Even(n).start();

}

else

{

new Odd(n).start();

}

}

}

}

class Even extends Thread

{

private int num;

Even(int n)

{

this.num = n;

}

public void run()

{

System.out.println("Square : "+(num\*num));

}

}

class Odd extends Thread

{

private int num;

Odd(int n)

{

this.num = n;

}

public void run()

{

System.out.println("Cube : "+(num\*num\*num));

}

}

public class multithread

{

public static void main(String[] args)

{

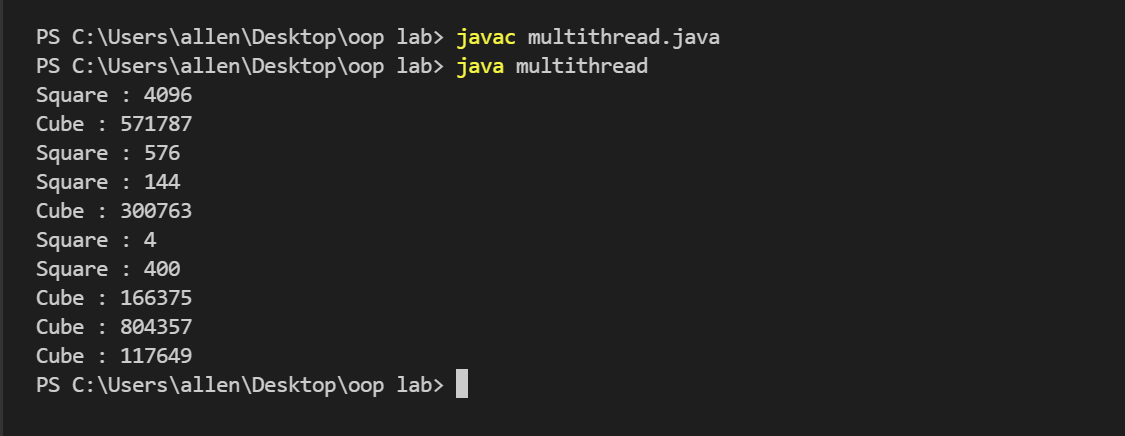
Randomizer rz = new Randomizer();

rz.start();

}

}

**OUTPUT :**

****

**RESULT :**

The program compiled and executed successfully.

**EXPERIMENT NO: 12**

**Date: 14/02/2022**

# AIM :

*To write a java program that shows thread synchronization.*

# ALGORITHM :

1. Start
2. Main function is invoked
   1. Object sc is created for SyncThread(“hello”)
   2. Object sd is created for SyncThread(“World”)
   3. Start method is called
3. Define class display
   1. In the method print
      1. In the try function do
         1. Sleep thread for 100 millisecond
      2. If any exception occurs catch the exception
         1. Print the message
4. Define SyncThread class which extends thread
   1. An object d is created for display
   2. SyncThread(String msg)
      1. This.msg = msg
   3. Method run is defined
      1. d.print(msg)
5. Stop

# PROGRAM CODE:

class Display

{

public void Print(String msg)

{

System.out.print("["+msg); try

{

Thread.sleep(100);

}

catch(InterruptedException e)

{

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

}

System.out.println("]");

}

}

class SyncThread extends Thread

{

Display d = new Display(); private String msg; SyncThread(String msg)

{

this.msg = msg;

}

public void run()

{

d.Print(msg);

}

}

public class synchronise

{

public static void main(String[] args)

{

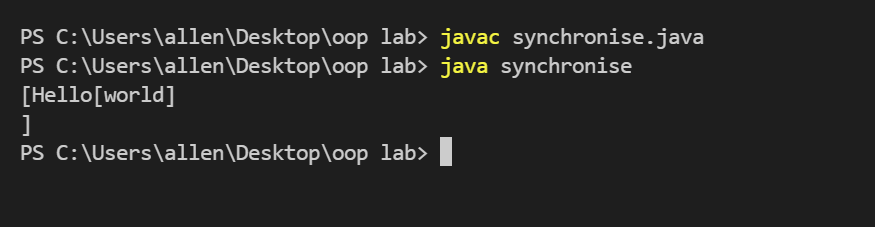
SyncThread sc = new SyncThread("Hello"); SyncThread sd = new SyncThread("world"); sc.start();

sd.start();

}

}

**OUTPUT :**

****

**RESULT :**

The program compiled and executed successfully.

**EXPERIMENT NO: 13**

**Date: 14/02/2022**

# AIM :

*To write a java program that works as a simple calculator. Arrange Buttons for digits and the + - \* % operations properly. Add a text field to display the result. Handle any possible exceptions like divide by zero.*

# ALGORITHM :

1. Start
2. Import packages
3. Class Calculator is created that implements interface ActionListener
4. Create an object of Calculator class and call its constructor
5. Create a new frame by creating object of JFrame
6. Declare necessary Jbutton and JtextField's and arrange them by setting bounds to them
7. Add the buttons to the frame
8. Add ActionListener to all the buttons
9. Method actionPerfomed is defined
   1. For every button pressed display its value in the textfield
   2. if (e.getSource()==button)
      1. tf.setText(tf.getText().concat("buttonName"))
   3. f the button pressed is an operator the store the current value of the textfield to an integer a
   4. if (e.getSource()==operatorButton)
      1. a=Double.parseDouble(tf.getText())
      2. op = 2
      3. tf.setText("")
   5. if equalto button is pressed then store the current value in textfield to integer variable b
   6. if (e.getSource()==EqualButton)
      1. b=Double.parseDouble(tf.getText())
   7. then perform operation using operator value stored in variable operator,on values a and b
10. switch(op)
    1. case 1:
       1. result=a+b
    2. case 2:
       1. result = a-b
    3. case 3:
       1. result = a\*b
    4. case 4:
       1. result = a/b 11.Display the result

12.If an exception is occurs catch the expression 13.Stop

# PROGRAM CODE:

import javax.swing.\*; import java.awt.event.\*; import java.awt.\*;

class Calculator implements ActionListener

{

JFrame f;

JTextField tf;

JButton b1,b2,b3,b4,b5,b6,b7,b8,b9,b0,bdot,badd,bsub,bmul,bdiv,beq,bdel,bclr; static double a=0,b=0,result=0;

static int op=0; Calculator()

{

f = new JFrame(); f.setTitle("Calculator"); f.setVisible(true); f.setLayout(null);

f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); f.setBounds(20,30,300,350);

f.setResizable(false);

Font font = new Font("Arial",Font.BOLD,20);

tf = new JTextField();

tf.setFont(font); tf.setBackground(Color.orange); tf.setForeground(Color.red); tf.setBounds(40,40,200,50); f.add(tf);

b1=new JButton("1"); b2=new JButton("2"); b3=new JButton("3"); b4=new JButton("4"); b5=new JButton("5"); b6=new JButton("6"); b7=new JButton("7"); b8=new JButton("8"); b9=new JButton("9"); b0=new JButton("0"); bdot = new JButton("."); badd=new JButton("+"); badd.setFont(font); bsub=new JButton("-"); bsub.setFont(font); bmul=new JButton("\*"); bmul.setFont(font); bdiv=new JButton("/"); bdiv.setFont(font); beq=new JButton("="); beq.setFont(font);

bdel=new JButton("Delete"); bclr=new JButton("clear"); b7.setBounds(40,100,50,40); b8.setBounds(90,100,50,40); b9.setBounds(140,100,50,40); bdiv.setBounds(190,100,50,40); b4.setBounds(40,140,50,40); b5.setBounds(90,140,50,40); b6.setBounds(140,140,50,40); bmul.setBounds(190,140,50,40); b1.setBounds(40,180,50,40); b2.setBounds(90,180,50,40); b3.setBounds(140,180,50,40); bsub.setBounds(190,180,50,40); bdot.setBounds(40,220,50,40); b0.setBounds(90,220,50,40); beq.setBounds(140,220,50,40); badd.setBounds(190,220,50,40); bdel.setBounds(40,260,100,40); bclr.setBounds(140,260,100,40); f.add(b1);

f.add(b2);

f.add(b3);

f.add(b4);

f.add(b5);

f.add(b6);

f.add(b7);

f.add(b8);

f.add(b9);

f.add(b0);

f.add(bdot);

f.add(badd);

f.add(bsub);

f.add(bmul);

f.add(bdiv);

f.add(beq);

f.add(bdel);

f.add(bclr); b1.addActionListener(this); b2.addActionListener(this); b3.addActionListener(this); b4.addActionListener(this); b5.addActionListener(this); b6.addActionListener(this); b7.addActionListener(this); b8.addActionListener(this); b9.addActionListener(this); b0.addActionListener(this); bdot.addActionListener(this); badd.addActionListener(this); bsub.addActionListener(this); bmul.addActionListener(this); bdiv.addActionListener(this); beq.addActionListener(this); bdel.addActionListener(this); bclr.addActionListener(this);

}

public static void main(String[] args)

{

new Calculator();

}

@Override

public void actionPerformed(ActionEvent e)

{

if (e.getSource()==b1)

{

tf.setText(tf.getText().concat("1"));

}

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

{

tf.setText(tf.getText().concat("3"));

}

else if (e.getSource()==b2)

{

tf.setText(tf.getText().concat("2"));

}

else if (e.getSource()==b4)

{

tf.setText(tf.getText().concat("4"));

}

else if (e.getSource()==b5)

{

tf.setText(tf.getText().concat("5"));

}

if (e.getSource()==b6)

{

tf.setText(tf.getText().concat("6"));

}

if (e.getSource()==b7)

{

tf.setText(tf.getText().concat("7"));

}

if (e.getSource()==b8)

{

tf.setText(tf.getText().concat("8"));

}

if (e.getSource()==b9)

{

tf.setText(tf.getText().concat("9"));

}

if (e.getSource()==b0)

{

tf.setText(tf.getText().concat("0"));

}

if (e.getSource()==bdot)

{

tf.setText(tf.getText().concat("."));

}

if (e.getSource()==badd)

{

a=Double.parseDouble(tf.getText()); op = 1;

tf.setText("");

}

if (e.getSource()==bsub)

{

a=Double.parseDouble(tf.getText()); op = 2;

tf.setText("");

}

if (e.getSource()==bmul)

{

a=Double.parseDouble(tf.getText()); op = 3;

tf.setText("");

}

if (e.getSource()==bdiv)

{

a=Double.parseDouble(tf.getText()); op = 4;

tf.setText("");

}

if (e.getSource()==bclr)

{

tf.setText("");

}

if (e.getSource()==bdel)

{

String s = tf.getText(); tf.setText("");

for(int i=0;i<s.length()-1;i++)

{

tf.setText(tf.getText()+s.charAt(i));

}

}

if (e.getSource()==beq)

{ int flag=0; b=Double.parseDouble(tf.getText());

switch(op)

{

case 1:

{

result=a+b; break;

}

case 2:

{

result = a-b; break;

}

case 3:

{

result = a\*b; break;

}

case 4:

{

if(b==0)

{

flag=1; tf.setText("MathError");

throw new ArithmeticException("Divide by Zero not Possible");

}

else

{

result=a/b;

}

break;

}

}

if (flag==0)

{

tf.setText(""+result);

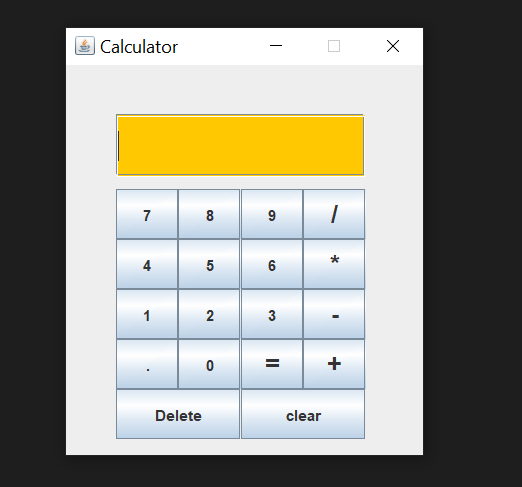
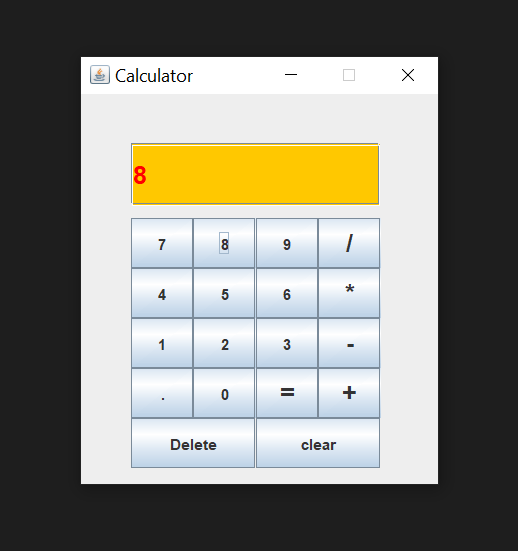
}

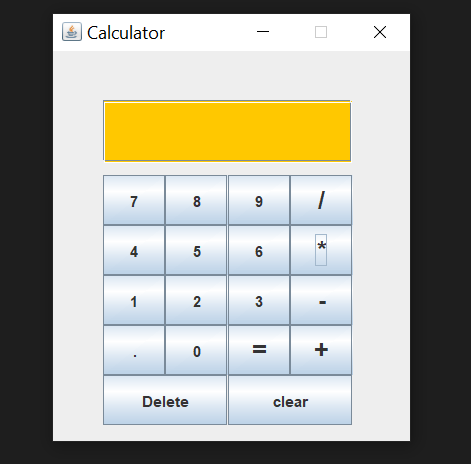
}

}

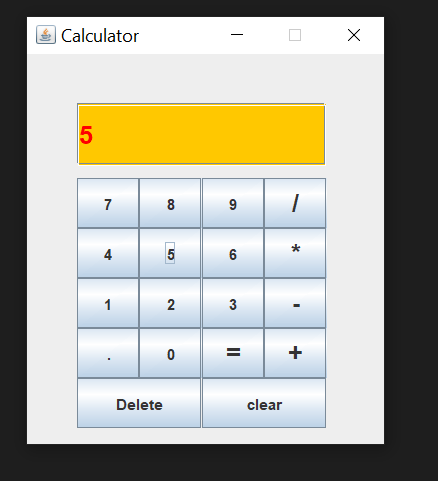
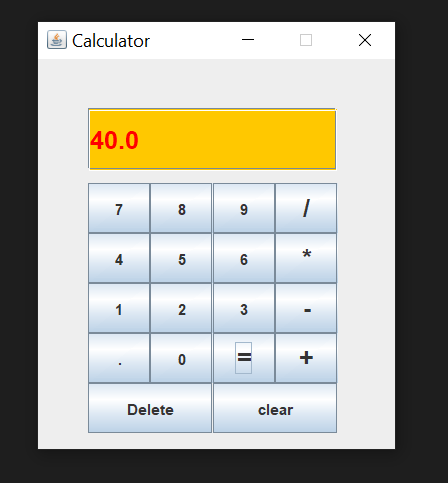
}

**OUTPUT :**

****

****

8 \* 5 = 40

****

**RESULT :**

The program compiled and executed successfully.

**EXPERIMENT NO: 14**

**Date: 21/02/2022**

# AIM :

*To write a java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time. No light is on when the program starts.*

# ALGORITHM :

1. Start
2. Import files java.awt and javax.swing
3. Main class trafficlight is created
   1. Object f is created for JFrame
   2. f.setVisible(true)
   3. Set size as 640 , 480
   4. f.setLayout(null)
   5. Object t is created for class light
   6. f.add(t)
4. Class light is created which extends JPanel and implements ActionListener
   1. Three radio buttons r1, r2 and r3 are created
   2. Three colors red\_c, yellow\_c and green\_c are created
   3. A constructor is created

i. setBounds(0,0,648,480)

* + 1. Give names to the radio buttons
    2. Make red color selected as default
    3. Group the buttons to select one button at a time
    4. Add r1, r2 and r3
    5. r1.addActionListener(this)
    6. r2.addActionListener(this)
    7. r3.addActionListener(this)
    8. Create a method actionPerfomed
       1. If r1 is selected
          1. red\_c = Color.red
          2. yellow\_c = getBackground()
          3. green\_c = getBackground()
       2. If r2 is selected
          1. red\_c = getBackground()
          2. yellow\_c = Color.yellow
          3. green\_c = getBackground()
       3. If r3 is selected

a. red\_c = getBackground()

b. yellow\_c = getBackground()

c. green\_c = Color.green

4. repaint function is called

i. Create a method paintComponent(Graphics g)

1. Super.paintComponent(g)

2. Draw three ovals with dimensions(50,50,50,50) ,

(50,110,50,50) and (50,170,50,50)

3. Set colors for each ovals

Stop

# PROGRAM CODE:

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

class light extends JPanel implements ActionListener

{

private JRadioButton r1; private JRadioButton r2; private JRadioButton r3;

private Color red\_c; private Color yellow\_c; private Color green\_c;

public light()

{

setBounds(0,0,640,480);

r1 = new JRadioButton("Red");

r2 = new JRadioButton("Yellow"); r3 = new JRadioButton("Green");

r1.setSelected(true); // to get red selected as default red\_c = Color.red;

yellow\_c = getBackground(); green\_c = getBackground();

ButtonGroup gp = new ButtonGroup();

gp.add(r1);

gp.add(r2);

gp.add(r3);

add(r1);

add(r2);

add(r3);

r1.addActionListener(this); r2.addActionListener(this); r3.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(r1.isSelected() == true)

{

red\_c = Color.red;

yellow\_c = getBackground(); green\_c = getBackground();

}

else if(r2.isSelected() == true)

{

red\_c = getBackground(); yellow\_c = Color.yellow; green\_c = getBackground();

}

else if(r3.isSelected() == true)

{

red\_c = getBackground(); yellow\_c = getBackground(); green\_c = Color.green;

}

repaint();

}

public void paintComponent(Graphics g)

{

super.paintComponent(g); g.drawOval(50,50,50,50); g.drawOval(50,110,50,50); g.drawOval(50,170,50,50);

g.setColor(red\_c); g.fillOval(50,50,50,50); g.setColor(yellow\_c); g.fillOval(50,110,50,50); g.setColor(green\_c); g.fillOval(50,170,50,50);

}

}

public class trafficlight

{

public static void main(String args[])

{

JFrame f = new JFrame();

f.setVisible(true);

f.setSize(640,480);

f.setLayout(null);

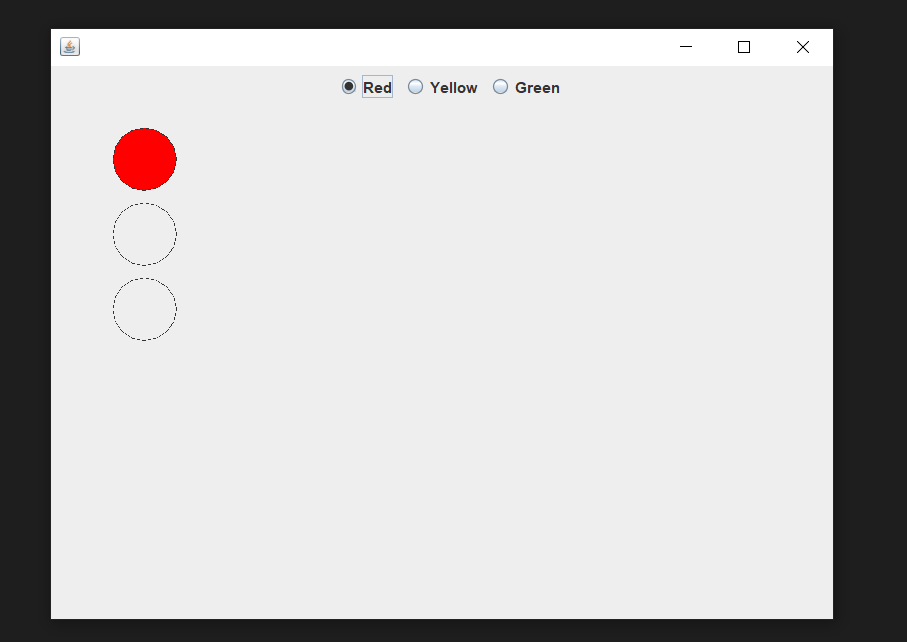
light t = new light();

f.add(t);

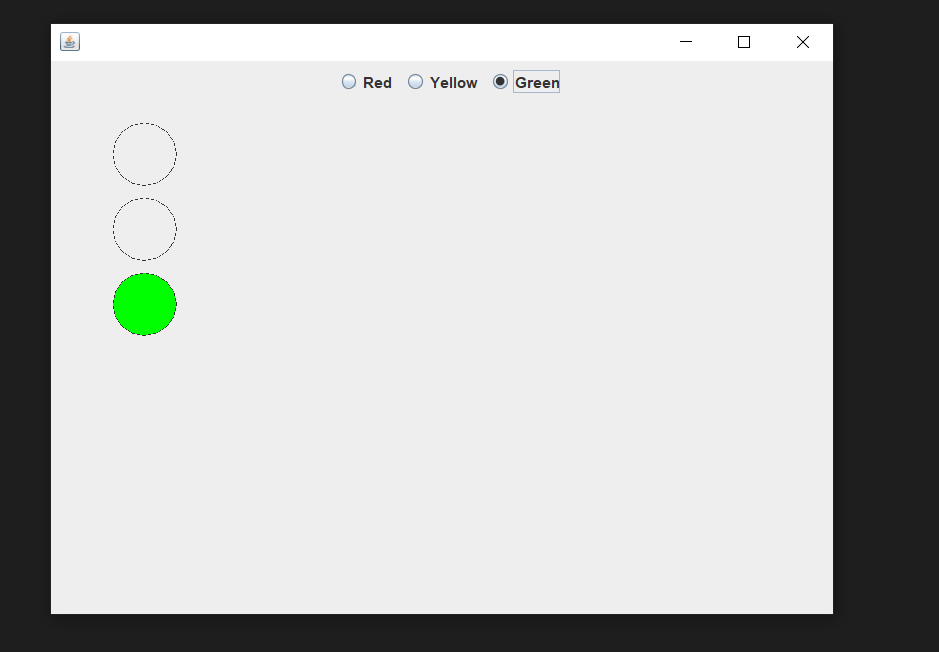
}

}

# OUTPUT :



****



**RESULT:**

The program compiled and executed succesfully.

**EXPERIMENT NO: 16**

**Date: 21/02/2022**

# AIM :

*To write a java program for the following :*

1. *Create a doubly linked list of elements.*
2. *Delete a given element from the above list.*
3. *Display the contents of the list after deletion*.

# ALGORITHM :

* 1. Start
  2. A class doubly linked list is defined
     1. Declare Node head, newnode, temp and ptr
     2. Node class is defined
        1. Declare data and pointers prev and link
        2. Using this keyword prev and link is assigned to null
     3. Insert function is created
        1. New node is created
        2. If head is null
           1. Head = new node
           2. Temp = new node
        3. Else
           1. Temp.link = new node
           2. Temp = new node
        4. Print insertion succesfull
     4. Delete function is created
        1. If head = null
           1. Print list is empty
        2. Else
           1. If head.link = null

Head = head.link

* + - * 1. Else

Ptr = head;

Head = head.link

Head.prev = null

* + 1. Display function is created
       1. Ptr = head
       2. While ptr != null
          1. Print ptr.data
          2. Prt = ptr.link
          3. Main class is invoked

An object dll is created for the class doubly linked list

Initialize choice variable as zero

While choice != 4

Switch choice

If input value is 1

A data is received from the user

Insert function is called

If input value is 2

Delete function is called

If input value is 3

Display function is called

If input value is 4

Exit from the program

* + - * 1. Stop

# PROGRAM CODE:

import java.util.Scanner;

class DoublyLinkedList

{

Node head = null;

class Node

{

int data;

Node next;

Node prev;

public Node(int data)

{

this.data = data;

this.next = null;

this.prev = null;

}

}

public void insert(int data)

{

Node temp = new Node(data);

if(head == null)

{

head = temp;

}

else

{

Node ptr = head;

while(ptr.next != null)

{

ptr = ptr.next;

}

ptr.next = temp;

temp.prev = ptr;

}

System.out.print("Data enterd is : "+data);

}

public void delete()

{

if(head == null)

{

System.out.print("List is empty");

}

else

{

int data = head.data;

head = head.next;

head.prev = null;

System.out.print("Data deleted is : "+data);

}

}

public void display()

{

Node temp = head;

if(head == null)

{

System.out.print("List is empty");

}

else

{

while(temp != null)

{

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

temp = temp.next;

}

}

}

}

public class linkedlist

{

public static void main(String args[])

{

DoublyLinkedList dll = new DoublyLinkedList();

int opt = 0;

while(opt != 4);

{

System.out.print("1. Inser at end");

System.out.print("\n2. Delete at front");

System.out.print("\n3. Display linked list");

System.out.print("\n4. Exit");

System.out.print("\nEnter your choice : ");

Scanner sc = new Scanner(System.in);

opt = sc.nextInt();

switch(opt)

{

case 1:

{

System.out.print("Enter the element : ");

int data = sc.nextInt();

dll.insert(data);

break;

}

case 2:

{

dll.delete();

break;

}

case 3:

{

dll.display();

break;

}

case 4:

{

break;

}

default:

{

System.out.print("Invalid choice");

break;

}

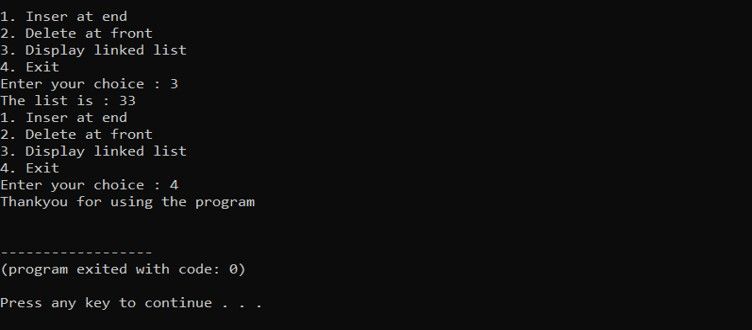
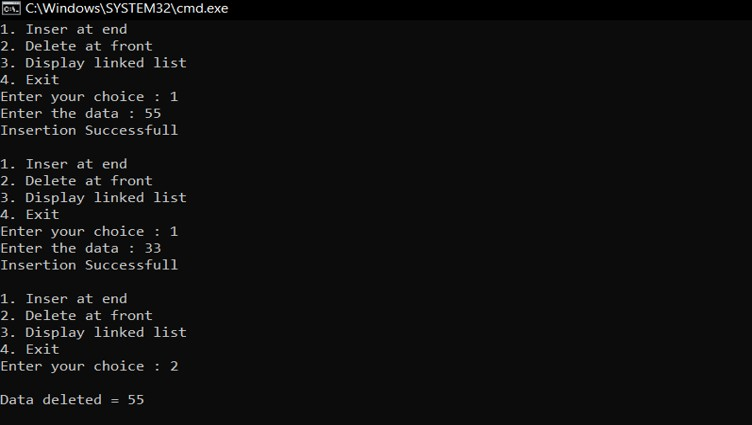
}

}

}

}

# OUTPUT :



**RESULT:**

The program compiled and executed successfully.

**EXPERIMENT NO: 17**

**Date: 28/02/2022**

# AIM :

*To write a java that implements Quick sort ALGORITHM for sorting a list of names in ascending order.*

# ALGORITHM :

1. Start
2. Create class quick sort
3. Quick sort function is invoked
   1. If p < r
      1. Set q = partition (A, p, r)
      2. Quick sort (A, p, q, -1)
      3. Quick sort (A, q, +1, r)
4. Partition function is invoked
   1. Assign string x = A[r] and int i = p - 1
   2. From j = p to j <= r – 1 do
      1. If A[j]. compareToIgnoreCase(x) <= 0 1. i = I + 1

2. temp = A[i] 3. A[i] = A[j]

1. A[j] = temp
   1. Temp = A[i + 1]

d. A[i + 1] = A[r]

1. A[r] = temp
2. Return i + 1
3. Main function is invoked
   1. Accept the limit value from the user as n
   2. For i = 0 to i < n do
      1. A[i] = sc.nextLine()
   3. Quick sort (A, 0, n, -1)
   4. Print the elements after quick sort
4. Stop

# PROGRAM CODE:

import java.util.Scanner;

class quicksort

{

public static void quickSort (String A[], int p, int r)

{

if (p < r)

{

int q = partition (A, p, r);

quickSort (A, p, q - 1);

quickSort (A, q + 1, r);

}

}

public static int partition (String A[], int p, int r)

{

String x = A[r];

int i = p - 1;

for (int j = p; j <= r - 1; j++)

{

if (A[j].compareToIgnoreCase(x) <= 0)

{

i = i + 1;

String temp = A[i];

A[i] = A[j];

A[j] = temp;

}

}

String temp = A[i + 1];

A[i + 1] = A[r];

A[r] = temp;

return i + 1;

}

public static void main(String args[])

{

Scanner sc = new Scanner (System.in);

System.out.print("--------------\n\n\n");

System.out.print("Enter the limit : ");

int n = sc.nextInt();

sc.nextLine();

String A[] = new String[n];

System.out.print("Enter the values : \n");

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

{

A[i] = sc.nextLine ();

}

quickSort (A, 0, n - 1);

System.out.println("\nAfter quick sort");

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

{

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

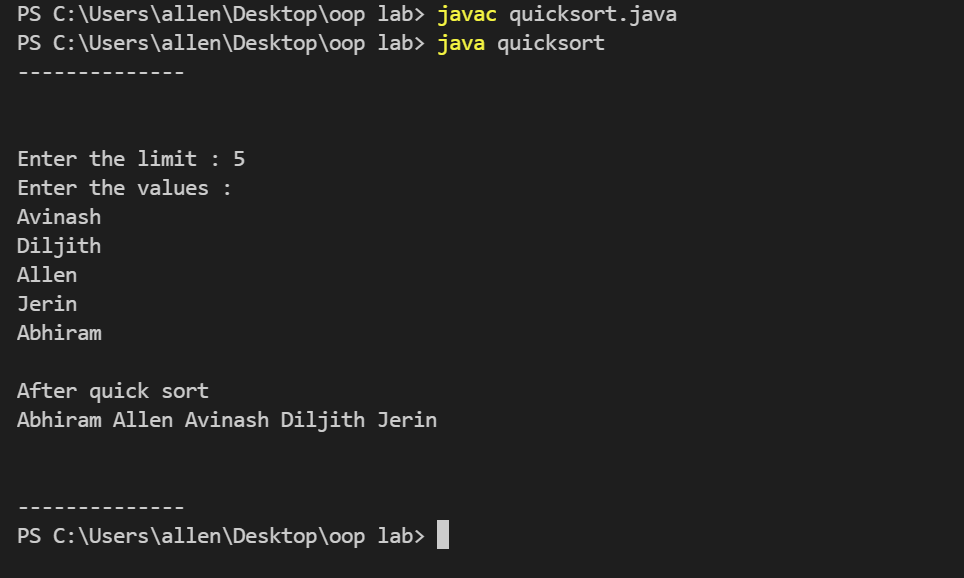
}

System.out.print("\n\n\n--------------");

}

}

**OUTPUT :**

****

**RESULT :**

The program compiled and executed successfully.

**EXPERIMENT NO: 18**

# Date: 28/02/2022

# AIM :

*To write a java program implements the binary search ALGORITHM.*

# ALGORITHM :

1. Start
2. Enter the number of elements
3. Read the array elements
4. Accept the search element
5. Method binary is called
6. Declare variables flag = 0 and index = 0
7. While low <= up
   1. mid=(low+up)/2
   2. if ar[mid] = key
      1. flag = 1
      2. index = mid + 1
   3. if ar[mid]<key
      1. low = mid +1
   4. if ar[mid] > key
      1. up = mid – 1
8. if flag == 1
   1. Print element found
9. Else
   1. Print element not found 10.Stop

# PROGRAM CODE:

import java.util.Scanner;

public class Binarysearch

{

static int ar[]=new int[50];

static void Binary(int low,int up,int key)

{

int mid;

int flag=0,index=0;

while (low<=up)

{

mid=(low+up)/2;

if (ar[mid]==key)

{

flag=1;

index = mid+1;

break;

}

else if (ar[mid]<key)

{

low=mid+1;

}

else if (ar[mid]>key)

{

up=mid-1;

}

}

if (flag==1)

{

System.out.println("Element found at position "+index);

}

else

{

System.out.println("Element not found");

}

}

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

int n,l,h;

System.out.print("--------------\n\n\n");

System.out.print("Enter the Number of Elements : ");

n = sc.nextInt();

l=0;h=n-1;

System.out.println("Enter "+n+" Elemnts to the array");

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

{

ar[i]=sc.nextInt();

}

System.out.print("Enter Search Element : ");

int key = sc.nextInt();

Binary(l,h,key);

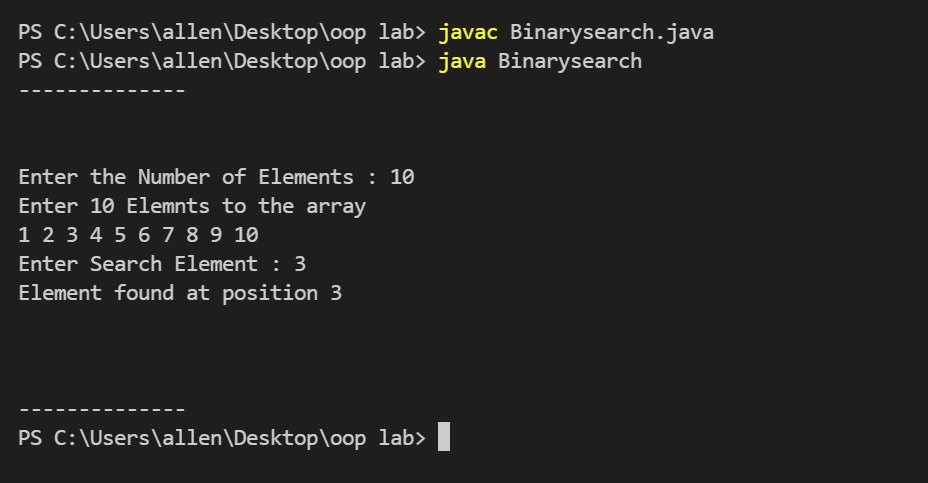
System.out.print("\n\n\n--------------");

sc.close();

}

}

**OUPUT :**

****

**RESULT :**

The program compiled and executed successfully.