DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE
(AUTONOMOUS)
(Approved by AICTE & Affiliated to Anna University, Chennai)
Accredited with 'A' Grade by NAAC, Accredited by TCS
Accredited by NBA with BME, ECE & EEE
PERAMBALUR - 621 212. Tamil Nadu.
website : www.dsengg.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

LAB MANUAL

U20IT505 JAVA PROGRAMMING LABORATORY

III YEAR / V SEMESTER

REGUALTION: 2020

Prepared By

AP / IT

Approved By

HOD / IT

LIST OF EXPERIMENTS

1. Prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate b2 -4ac is negative, display a message stating that there are no real solutions.
2. The Fibonacci sequence is defined by the following rule: The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a Java program that uses both recursive and non recursive functions to print the nth value in the Fibonacci sequence.
3. Prompts the user for an integer and then prints out all prime numbers up to that integer. (use Scanner class to read input)
4. Multiply two given matrices.
5. Reads a line of integers, and then displays each integer, and the sum of all the integers (Use String Tokenizer class of java.util)
6. Checks whether a given string is a palindrome or not. Ex: MADAM is a palindrome.
7. Sorting list of names. Read input from command line.
8. Make frequency count of words in a given text.
9. Program to create a Student class with following fields i.Hall ticket number ii.Student Name iii. Department Create ‘n’ number of Student objects where ‘n’ value is passed as input to Constructor.
10. Demonstrate String comparison using == and equals method.

TOTAL: 60 PERIODS

BIBLIOGRAPHY

TEXT/REFERECE BOOKS:

1. Herbert Schildt, “JavaTM : The Complete Reference”, 9th edition, Oracle Press, 2014.
2. Anita Seth, B. L. Juneja, “JAVA: One Step Ahead”, Oxford University Press, 2017.
3. Cay S. Horstmann and Gary Cornell, “Core Java: Volume I Fundamentals”, 9th edition, Prentice Hall, 2013.
4. K. Arnold, D. Holmes and J. Gosling, “The JAVA programming language”, 4th edition, Addison Wesley Professional, 2005.
5. Timothy Budd, “Understanding Object-oriented programming with Java”, 3rd edition, AddisonWesley, 2000.

LIST OF ADDITIONAL EXERCISE

1. Write the Java Program to Check Whether a Number is Even or Odd.
2. Write the Java Program to Check Leap Year.
3. Write the Java Program to Add Two Integers.
4. Write the Java Program to print prime numbers from 1 to N in reverse order.
5. Calculate the Java Program to Add Two Matrix Using Multi-dimensional Arrays.
6. Write the Java Program to Illustrate Methods of StringTokenizer class (hasMoreToken(), next Token() and countTokens()).
7. Implement the Java Program to Add Two Complex Numbers by Passing Class to a Function.
8. Implement the Java Program to sort the elements of an array in descending order.
9. Write the java program to hasMoreTokens () method of the StringTokenizer class.
10. Write the Java program to get Employee details using string concept.
11. Compute the Java Program to create a Calculator by an Applet.

EX.NO: 01 DATE:	WRITE A JAVA PROGRAM THAT PRINTS ALL REAL SOLUTIONS TO THE QUADRATIC EQUATION $AX^2+BX+C = 0$. READ IN A, B, C AND USE THE QUADRATIC FORMULA. IF THE DISCRIMINATE B^2-4AC IS

AIM:

To write a Java programthat prints all real solutions to the quadratic equation $ax^2+bx+c = 0$.

ALGORITHM:

Step 1: Start

Step 2: Read a, b, c

Step 3: initialize d<-(b*b)-(4*a*c)

Step 4: initialize r<- b/2*a

Step 5: if d>0 go to Step 6, else go to Step 8

Step 6: r1=r+(sqrt(d)/2*a) and r2=r-(sqrt(d)/2*a)

Step 7: prints roots are real and distinct, first root r1 second root r2

Step 8: if d=0 go to Step 9, else go to Step 10

Step 9: print roots are real and equal, -r

Step 10: d=-d

Step 11: im=sqrt(d)/2*a

Step 12: print roots are imaginary, first root is r+Iim, and the second root is r-Iim

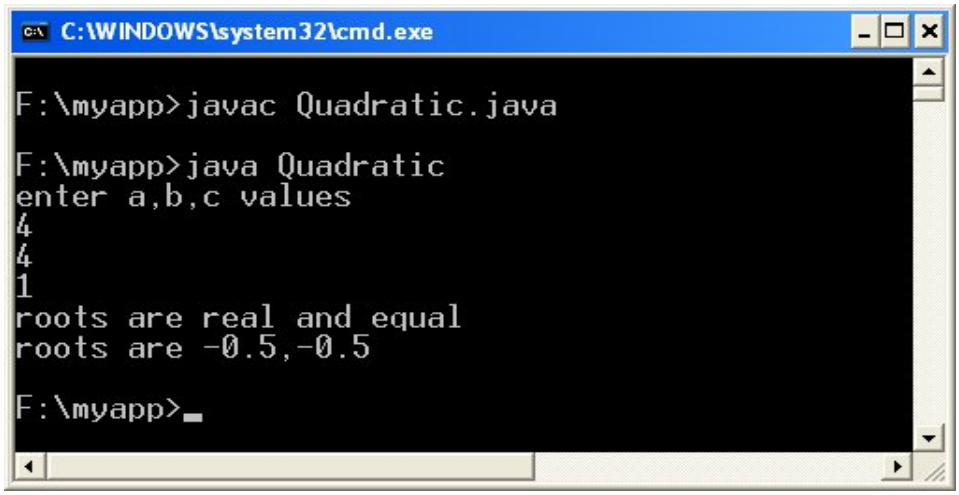
Step 13: Stop

SAMPLE PROGRAM :-

```
import java.io.*;
class quadratic
{
        public static void main(String args[])throws IOException
        {
                double x1,x2,disc,a,b,c;
                InputStreamReader obj=new InputStreamReader(System.in);
                BufferedReader br=new BufferedReader(obj);
                System.out.println("enter a,b,c values");
                a=Double.parseDouble(br.readLine());
                b=Double.parseDouble(br.readLine());
                c=Double.parseDouble(br.readLine());
                disc=(b*b)-(4*a*c);
```

```
                        if(disc==0)
                        {
                                System.out.println("roots are real and equal ");
                                x1=x2=-b/(2*a);
                                System.out.println("roots are "+x1+","+x2);
                        }
                        else if(disc>0)
                        {
                                System.out.println("roots are real and unequal");
                                x1=(-b+Math.sqrt(disc))/(2*a);
                                x2=(-b+Math.sqrt(disc))/(2*a);
                                System.out.println("roots are "+x1+","+x2);
                        }
                        else
                        {
                                System.out.println("roots are imaginary");
                        }
                }
        }
```

INPUT & OUTPUT :-



EXERCISE PROGRAM:

1. Write the Java Program to Check Whether a Number is Even or Odd.

RESULT:

Thus the above program has been executed and verified successfully.

EX.NO: 02 DATE:	THE FIBONACCI SEQUENCE IS DEFINED BY THE FOLLOWING RULE. THE FIRST 2 VALUES IN THE SEQUENCE ARE 1, 1. EVERY SUBSEQUENT VALUE IS THE SUM OF THE 2 VALUES PRECEDING IT. WRITE A JAVA PROGRAM THAT USES BOTH RECURSIVE AND NON-RECURSIVE

AIM:

To write a Java program that uses both recursive and non-recursive functions to print the n^{th} value of the Fibonacci sequence.

ALGORITHM:

Step 1: Start

Step 2: Declare variables I, a,b , show

Step 3: Initialize the variables, a=0, b=1, and show =0

Step 4: Enter the number of terms of Fibonacci series to be printed

Step 5: Print First two terms of series

Step 6: Use loop for the following steps

-> show=a+b

-> a=b

-> b=show

-> increase value of I each time by 1

-> print the value of show

Step 7: End

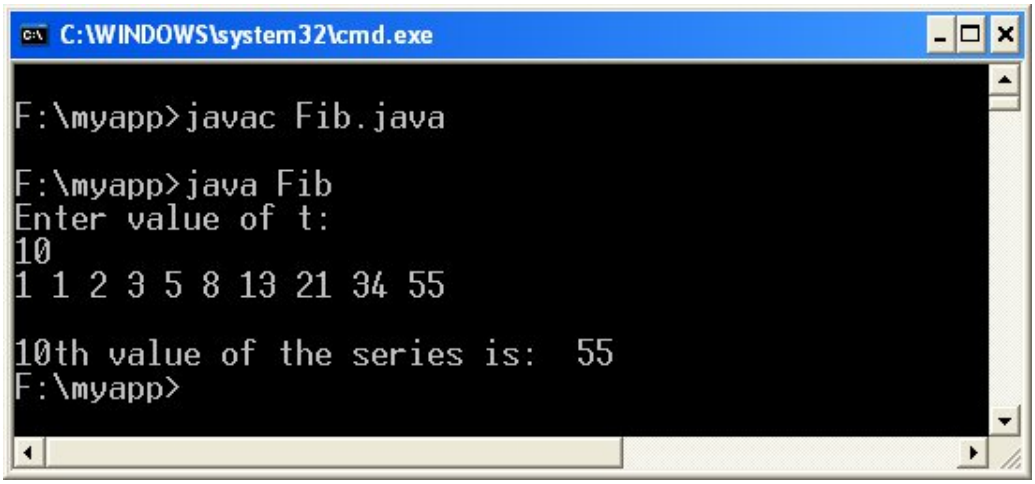
SAMPLE PROGRAM:-

/*Non Recursive Solution*/

```
import java.util.Scanner;
class Fib
{
	public static void main(String args[])
		{
			Scanner input=new Scanner(System.in);
			int i ,a=1,b=1,c=0,t;
			System.out.println("Enter value of t:");
			t=input.nextInt();
			System.out.print(a);
			System.out.print(""+b);
```

```
                    for(i=0;i<t-2;i++)
                    {
                        c=a+b; a=b; b=c;
                        System.out.print(" "+c);
                    }

                System.out.println();
                System.out.print(t+"th value of the series is: "+c);

            }

}
```

INPUT & OUTPUT:-



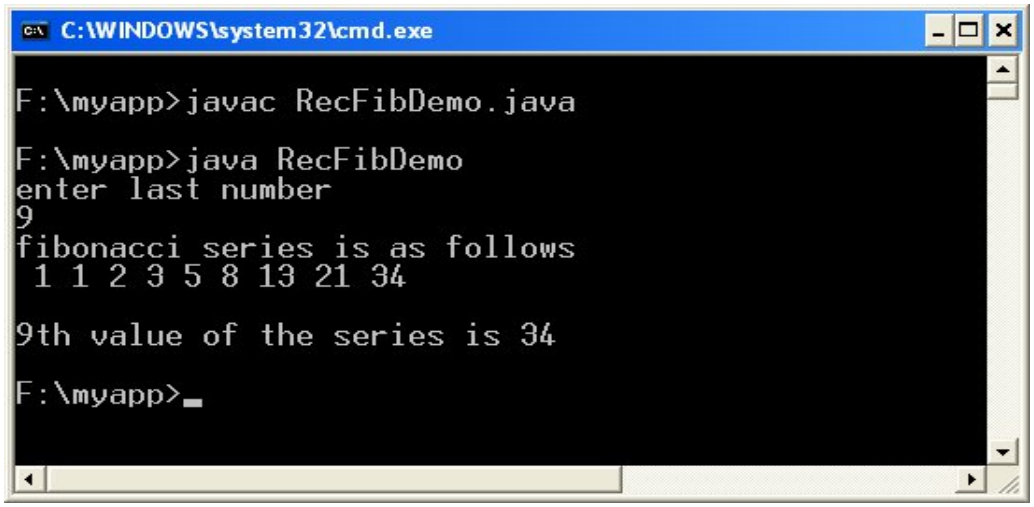
/* Recursive Solution*/

```
import java.io.*;

 import java.lang.*;

class Demo
{

            int fib(int n)
             {
                    if(n==1)

                     {

                       return (1);

                     }
```

```
                    else if(n==2)
                    {
                        return (1);
                    }
                    else
                    {
                      return (fib(n-1)+fib(n-2));
                    }
}
class RecFib Demo
{
        public static void main(String args[])throws IOException
          {
                InputStreamReader obj=new InputStreamReader(System.in);
                BufferedReader br=new BufferedReader(obj);
                System.out.println("enter last number");
                int n=Integer.parseInt(br.readLine());
                Demo ob=new Demo();
                System.out.println("fibonacci series is as follows");
                int res=0;
                    for(inti=1;i<=n;i++)
                    {
                        res=ob.fib(i);
                        System.out.println(" "+res);
                    }
                  System.out.println();
                  System.out.println(n+"th value of the series is "+res);
          }
```

}

INPUT & OUTPUT :-



EXERCISE PROGRAM:

1. Write the Java Program to Check Leap Year.
2. Write the Java Program to Arithmetic operation of Two Integers.

RESULT:

Thus the above program has been executed and verified successfully.

EX.NO: 03 DATE:	A PROMPTS THE USER FOR AN INTEGER AND THEN PRINTS OUT ALL THE PRIME NUMBERS UP TO THAT INTEGER.

AIM:

To write a Java program prints out all the prime numbers up to that Integer.

ALGORITHM:

Step 1: Start of Program

Step 2: Take two integer n and p.

Step 3: Take n as user input.

Step 4: Use for loops to check all the Prime number between 2-n.

Step 5: If n is Prime print n.

Step 6: If n is not prime, skip.

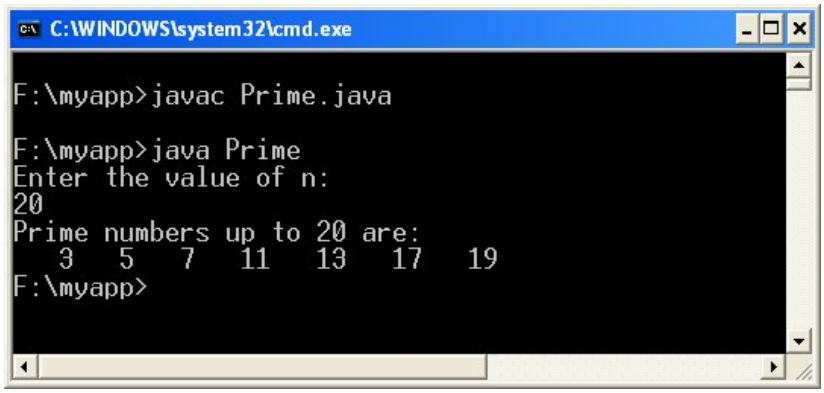
Step 7: End of Program

SAMPLE PROGRAM:-

```
import java.util.*;

class Test
 {

        void check(int num)
        {
            System.out.println("Prime numbers up to "+num+" are:");
                for(int i=1;i<=num;i++)
                {
                    for(int j=2;j<i;j++)
                    {
                        if(i%j==0)
                        {
                            break;
                        }

                        else if((i%j!=0)&&(j==i-1))
                        {
                            System.out.print(' '+i);
                        }
                    }

                }
        }
```

```
 }                                          //end of class Test class Prime

public static void main(String args[ ])
{
        Test obj1=new Test();
        Scanner input=new Scanner(System.in);
        System.out.println("Enter the value of n:");
        int n=input.nextInt();
        obj1.check(n);
}
```

INPUT & OUTPUT :-



EXERCISE PROGRAM:

1. Write the Java Program to Print prime numbers from 1 to N in reverse order.
2. Write the java program to String operations(copy(), reverse(), concatenation(), length()

RESULT:

Thus the above program has been executed and verified successfully.

EX.NO.: 04 DATE:	TO MULTIPLY TWO GIVEN MATRICES.

AIM:

To write a Java program to multiply two given matrices.

ALGORITHM:

Step 1: Start

Step 2: Declare variables for matrix size.

Step 3: Initialize the number of rows and

columns for the first matrix.

Step 4: Initialize the number of rows and

columns for the second matrix.

Step 5: Declare two matrices.

Step 6: Ask the user to initialize the matrices.

Step 7: Call a method to multiply the two matrices.

Step 8: Print the two matrices.

Step 9: Check if matrix multiplication is possible or not.

Step 10: If possible, then create a new Matrix to store the product of the two matrices.

Step 12: Traverse each element of the two matrices and multiply them.

Step 13: Store this product in the new matrix at the corresponding index.

Step 14: Print the final product matrix.

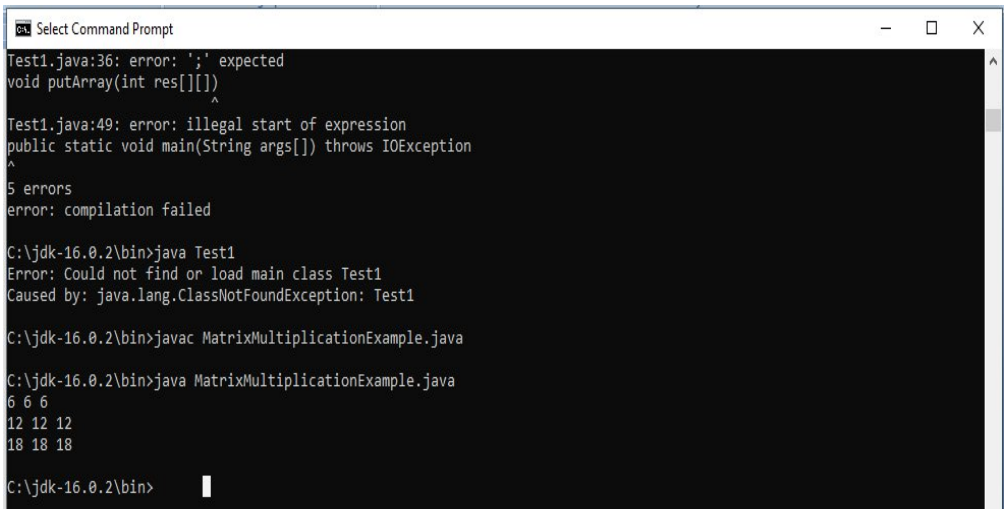
Step 15: If matrix multiplication is not possible, then display the same.

Step 16: Stop.

SAMPLE PROGRAM :-

```
public class MatrixMultiplicationExample
{
        public static void main(String args[])
        {
                int a[][]={{1,1,1},{2,2,2},{3,3,3}};
                int b[][]={{1,1,1},{2,2,2},{3,3,3}};
                 int c[][]=new int[3][3];
                        for(int i=0;i<3;i++)
                        {
                            for(int j=0;j<3;j++)
                            {
                            c[i][j]=0;
                                for(int k=0;k<3;k++)
                                {
                                c[i][j]+=a[i][k]*b[k][j];
                                }
                            System.out.print(c[i][j]+" ");
                            }
                        System.out.println();
                        }
        }
}
```

INPUT & OUTPUT :-



EXERCISE PROGRAM:

1. Java Program to Add Two Matrix Using Multi-dimensional Arrays.
2. Java Program to Transpose Matrix.

RESULT:

Thus the above program has been executed and verified successfully.

EX.NO: 05 DATE:	READS A LINE OF INTEGERS AND THEN DISPLAYS EACH INTEGER AND THE SUM OF ALL INTEGERS. (USE STRINGTOKENIZER CLASS).

AIM:

To Write a Java program that reads a line of integers and then displays each integer and the sum of all integers. (Use StringTokenizer class).

ALGORITHM:

Step 1: Import java.util.Scanner

Step 2: Sum = 0

Step 3: Initialise Scanner object in on System.in

Step 4: S = in.nextLine() //read the numbers separated by a space

Step 5: InitialiseStringTokenizer object str on s

Step 6: While(str.hasMoreTokens())

Step 7: Num = Integer.parseInt(str.nextToken())

Step 8: Print num

Step 9: Sum += num

Step 10: Endwhile

Step 11: Print sum

PROGRAM :-

```
// Using StringTokenizer class

class StringTokenizerDemo

{

        public static void main(String args[])

        {

            int n;

            int sum = 0;
```

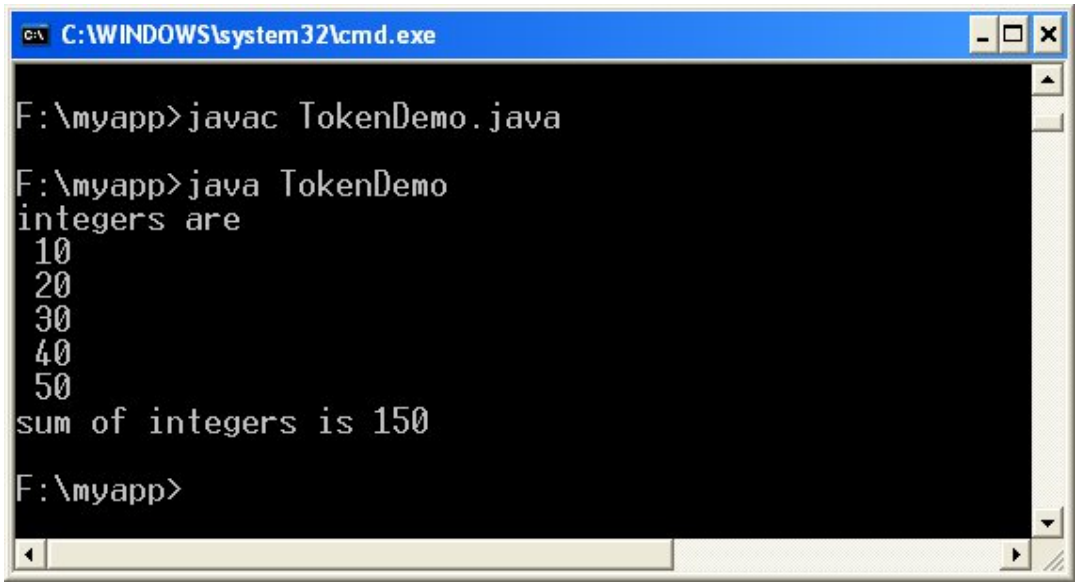
```
                    Scanner sc = new Scanner(System.in);
                    System.out.println("Enter integers with one space gap:");
                    String s = sc.nextLine();
                    StringTokenizer st = new StringTokenizer(s, " ");
                        while (st.hasMoreTokens())
                        {
                            String temp = st.nextToken();
                            n = Integer.parseInt(temp);
                            System.out.println(n);
                            sum = sum + n;
                        }
                    System.out.println("sum of the integers is: " + sum);
                    sc.close();
                }
}
```

// Alternate solution using command line arguments

```
  class Arguments
  {
        public static void main(String args[ ])
        {
              int sum=0;
              int n=args.length;
              System.out.println("length is "+n);
              intarr[]=new int[n];
```

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

        arr[i]=Integer.parseInt(args[i]);

        System.out.println("The enterd values are:");

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

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

        System.out.println("sum of enterd integers is:");

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

        sum=sum+arr[i];

        System.out.println(sum);

    }

}
```

INPUT & OUTPUT



SAMPLE PROGRAM:

1. Write the Java Program to Illustrate Methods of StringTokenizer class (hasMoreToken(), nextToken() and countTokens())

RESULT: Thus the above program has been executed and verified successfully.

EX.NO.: 06 DATE:	CHECKS WHETHER A GIVEN STRING IS A PALINDROME OR NOT. EX: MADAM IS A PALINDROME.

AIM:

To write a Java program that checks whether a given string is a palindrome or not.

ALGORITHM:

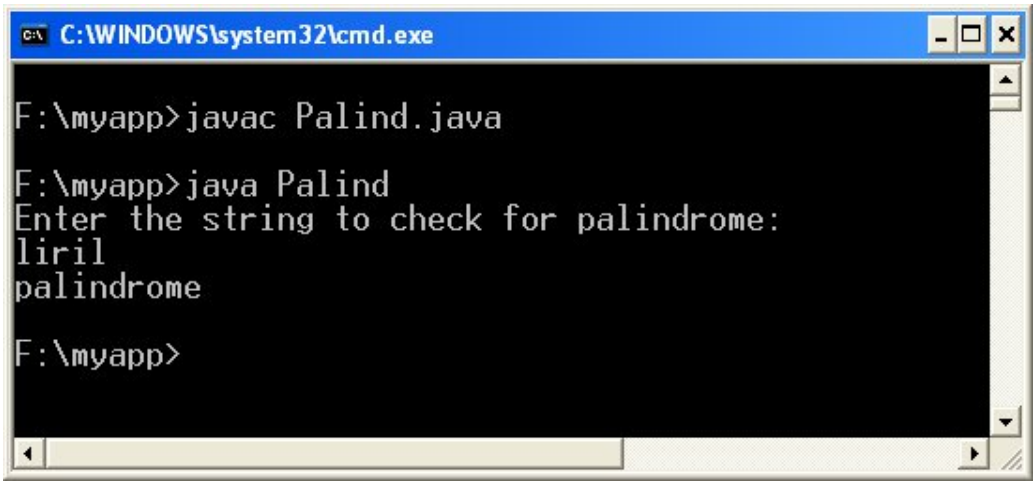
Step 1: Start
Step 2: Input the number.
Step 3: Find the reverse of the number.
Step 4: If the reverse of the number is equal to the number, then return true.
Step 5: Else, return false.
Step 6: Stop.

PROGRAM:-

```
import java.io.*; class Palind

{

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

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

            System.out.println("Enter the string to check for palindrome:");

            String  s1=br.readLine(); StringBuffer sb=new StringBuffer();

            sb.append(s1);

            sb.reverse();

            String s2=sb.toString();

                    if(s1.equals(s2))

                    {

                        System.out.println("palindrome");
                    }
                    else
                    {
                        System.out.println("not palindrome");
                    }
```

```
        }
}
```

INPUT & OUTPUT :-



SAMPLE PROGRAM:

1. To implement the Java Program to Add Two Complex Numbers by Passing Class to a Function.

RESULT:

Thus the above program has been executed and verified successfully.

EX.NO: 07 DATE:	A SORTING A GIVEN LIST OF NAMES IN ASCENDING ORDER.

AIM:

To write a Java program that for sorting a given list of names in ascending order.

ALGORITHM:

Step 1: Start
Step 2: Create a class name ex2 and declare the variables int n and string s[]
Step 3: Allocate the memory of names for variable s.
Step 4: Get total number of names to be entered in the variable n.
Step 5: Using compare To function sort the names in alphabetical order.
Step 6: Use for loop to sort the name.
Step 7: Print the sorted name list.
Step 8: Stop.

PROGRAM:-

```
import java.util.*;
class Sorting
{
	void sortStrings()
	{
		Scanner s = new Scanner(System.in);
		System.out.println("Enter the value of n: ");
		int n = s.nextInt();
		String[] str = new String[n];
		System.out.println("Enter strings: ");
		for(int i = 0; i < n; i++)
		{
			str[i] = new String(s.next());
		}
		for(int i = 0; i < n; i++)
		{
			for(int j = i+1; j < n; j++)
			{
				if(str[i].compareTo(str[j])>0)
				{
					String temp = str[i];
					str[i] = str[j];
					str[j] = temp;
```

```
                        }
                    }
                }
                System.out.println("Sorted list of strings is:");
                for(int i = 0; i < n ; i++)
                {
                    System.out.println(str[i]);
                }
        }
}
class Driver
{
        public static void main(String[] args)
        {
                Sorting obj = new Sorting();
                obj.sortStrings();
        }
}
```

INPUT & OUTPUT :-

```
Enter the value of n:
4
Enter strings:
suresh
mahesh
dinesh
ganesh
Sorted list of strings is:
dinesh
ganesh
mahesh
suresh
```

SAMPLE PROGRAM:

1. To implement the Java Program to sort the elements of an array in descending order.

RESULT:

Thus the above program has been executed and verified successfully.

EX.NO: 08 DATE:	THE READS A LINE OF INTEGERS AND THEN DISPLAYS EACH INTEGER AND THE SUM OF ALL INTEGERS. (USE STRINGTOKENIZER CLASS).

AIM:

To write Java program that make frequency count of words in a given text.

ALGORITHM:

Step 1: Define a string.

Step 2: Define an array freq with the same size of the string.

Step 3: Two loops will be used to count the frequency of each character. Outer loop will be used to select a character and initialize element at corresponding index in array freq with 1.

Step 4: Inner loop will compare the selected character with rest of the characters present in the string.

Step 5: If a match found, increment element in freq by 1 and set the duplicates of selected character by '0' to mark them as visited.

Step 6: Finally, display the character and their corresponding frequencies by iterating through the array freq.

PROGRAM:

```
public class Frequency
{
        public static void main(String[] args)
        {
            String str = "picture perfect";
            int[] freq = new int[str.length()];
            int i, j;
            char string[] = str.toCharArray();
                for(i = 0; i<str.length(); i++)
                {
                freq[i] = 1;
                    for(j = i+1; j <str.length(); j++)
                        {
```

```
                    if(string[i] == string[j])
                    {
                            freq[i]++;
                            string[j] ='0';
                    }
                    }
                }
                System.out.println("Characters and their corresponding frequencies");
                for(i = 0; i<freq.length; i++)
                {
                    if((string[i]!=' ')&&(string[i]!='0'))
                    System.out.println(string[i] +" "+ freq[i]);
                }
            }
}
```

OUTPUT:

```
Characters and their corresponding frequencies
p-2
i-1
c-2
t-2
u-1
r-2
e-3
f-1
```

SAMPLE PROGRAM:

1. Write the java program to hasMoreTokens () method of the StringTokenizer class.

RESULT:

Thus the above program has executed and verified successfully.

EX.NO: 09 DATE:	JAVA PROGRAM TO GET STUDENT DETAILS USING STRING CONCEPT.

AIM:

To write a Java program to get student details using String Concept.

ALGORITHM:

Step 1: Start

Step 2: Declare Names of the students.

Step 3: Declare Roll no of the students.

Step 4: Declare Marks of the students.

Step 5: Get the student mark details

Step 6: Stop

SAMPLE PROGRAM:

```
import java.util.*;
class Student
{
        String name;
        int rno;
        int marks;
          Student(String n,int r,int m)
          {
             this.name = n;
             this.rno =r;
             this.marks =m;
          }
          public  void  display()
          {
                System.out.println("NAME:"+this.name);
                System.out.println("RNO:"+this.rno);
                System.out.println("MARKS:"+this.marks);
```

```
        }
}
class Sampleclass
{
        public static void main(String[] args)
        {
            List<Student>lists = new ArrayList<Student>();
            String[] na={"Arun","Raj","Veera"};
            int[] rn ={1,2,3};
            int[] ma ={80,90,100};
              for(int i=0;i<=3;i++)
               {
                  Student s=new Student(na[i],rn[i],ma[i]);
                  lists.add(s);
               }
                    for(int i=0;i<=3;i++)
                      {
                          lists.get(i).display();
                       }
        }
}
```

Program2:

```
import java.io.*;
class Student_Details
{
        String name;
        int roll;
        String cname;
        String uname;
         InputStreamReader in = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(in);
```

```
        void input() throws IOException
        {
            System.out.print("Enter Student Name: ");
            name = br.readLine();
            System.out.print("Enter Roll Number: ");
            roll = Integer.parseInt(br.readLine());
            System.out.print("Enter College Name: ");
            cname = br.readLine();
            System.out.print("Enter University Name: ");
            uname = br.readLine();
        }
}
        class Main extends Student_Details
        {
                void display()
                {
                    System.out.print("\n**** Student Details ****\n" );
                    System.out.print("\nStudent Name: " + name);
                    System.out.print("\nRoll Number: " + roll);
                    System.out.print("\nCollege Name: " + cname);
                    System.out.print("\nUniversity Name: " + uname);
                }
        }
public static void main(String args[]) throws IOException
{
        Main c = new Main();
        c.input();
        c.display();
}
```

OUTPUT:

NAME: Arun

RNO: 1

MARKS: 80

NAME: Raj

RNO: 2

MARKS: 90

NAME: Veera

RNO: 3

MARKS: 100

SAMPLE PROGRAM:

1. To Write the Java program to get Employee details using string concept.

RESULT:

Thus the above program has been executed and verified successfully.

EX.NO: 10 DATE:	DEMONSTRATE STRING COMPARISON USING == AND EQUALS METHOD.

AIM:

To write a Java program to demonstrate String comparison using == and equals method.

ALGORITHM:

Step 1: Start

Step 2: Declare two strings

Step 3: Initialize them.

Step 4: Use the == operator to check whether the two strings are equal or not.

Step 5: Print the output.

Step 6: Stop.

PROGRAM:

```
public class Main1
{
        public static void main(String args[])
        {
                String s1 ="Study tonight";
                System.out.println("The entered string is "+s1);
                String s2="Study tonight";
                System.out.println("The entered string is "+s2);
                System.out.println("Are the two strings equal?");
                //Compare the strings using == operator
                        if(s1==s2)
                        {
                            System.out.println("Yes the entered strings are equal");
                        }
                        else
                        {
                            System.out.println("No the entered strings are not equal");
```

```
                    }
        }
  }
```

OUTPUT:

```
The entered string is Study tonight

The entered string is Study tonight

Are the two strings equal?

Yes the entered strings are equal
```

SAMPLE PROGRAM:

1. Write the Java Program to create a Calculator by an Applet

RESULT:

Thus the above program has been executed and verified successfully.