OBJECT ORIENTED PROGRAMMING USING JAVA LAB(CSL 203)

PROGRAM 1

AIM

Write a java program to check whether a given number is prime or not.

ALGORITHM

Step 1: Start

Step 2: initialising num=31,m=num/2

Step 3: set flg=0

Step 4: if number is 0 or 1 then it is neither prime nor composite

Step 5: For i=2 to m

Step 6: If n mod i=0 then the number is not prime

Step 8: else the number is prime

Step 9: Stop

PROGRAM

```
public class prime {
  public static void main(String[] args) {
      int num=31;
      int i;
int m=num/2;
int flg=0;
if(num==1|num==0)
{
```

```
System.out.println(num+"  neither prime nor composite");


}


else{
   for(i=2;i<=m;i++)
   if(num%i==0)
   { flg=-1;
   System.out.println(num+"is not a prime");
   break;
   }


if(flg==0)
System.out.println(" it is a prime");
}
   }
}
```

RESULT

it is a prime

EXPERIMENT NO 2

AIM

Write a java program to find the second smallest element in an array.

ALGORITHM

Step 1: start

Step 2: define a class secondsmall

Step 3: declare temp

Step 4: initialise an 1d array

Step 5: using nested for loop find second smallest element

Step 6: print the second smallest element

Step 7: stop

PROGRAM

```
public class secondsmall {
 public static void main(String[] args) {
   int temp;
   int i;
   int j;
   int a[]={3,0,-1,0,-2,-2,2,1,4};
   int n=a.length;
   for( i=0;i<n;i++)
   for( j=i+1;j<n;j++)
   {
      if(a[i]>a[j])
      {
```

```
            temp=a[i];
            a[i]=a[j];
            a[j]=temp;
          }
       }
       int p=0;
       int q=p+1;
       if(a[p]!=a[q])
        System.out.println("second smallest element is "+ a[1]);
else{
int c=0;
       for( i=0;i<n;i++)
      {  if(a[0]==a[i])
          c++;
      }
      System.out.println("second smallest element is "+ a[c]);
   }
}
}
```

RESULT

second smallest element is -1

EXPERIMENT NO 3

AIM

Write a java program to find the frequency of a given character in a string.

ALGORITHM

Step 1: start

Step 2: initialize a string and the character ,whose frequency has to be calculated

Step 3: store the string as an character array

Step 4: find the frequency of the given character using for loop

Step 5: print the result

Step 6: stop

PROGRAM

```
public class string{
public static void main(String []args)
   {
int c=0;
int i;
char ch='O';
char chr='o';
       String str="Welcome to the world Of programming";
       int n=str.length();
       char string[] = str.toCharArray();
       for(i=0;i<n;i++)
       {
  if(ch==string[i]||chr==string[i])
     c++;
```

```
    }
    System.out.println(" frequency of " + ch + "  in welcome to the world of programming is
"+ c);
  }
```

RESULT

frequency of O in welcome to the world of programming is 5

EXPERIMENT NO 4

AIM

Write a java program to reverse a given string.

ALGORITHM

Step 1: start

Step 2: enter a string

Step 3: find its length using . operator and store the string as a character array

Step 4: reverse the string using for loop

Step 5: print the reverse of string

Step 6: stop

PROGRAM

```
public class reverse {
  public static void main(String []args)
  {
    String str="welcome to the world of programming";
    int n=str.length();
    char string[] = str.toCharArray();
```

```
        for(int i=n-1;i>=0;i--)
        System.out.print(string[i]);
    }
}
```

RESULT

gnimmargorp fo dlrow eht ot emoclew

EXPERIMENT NO:5

AIM

To write a Java program to multiply two given matrices

ALGORITHM:

Step 1: Start

Step 2: Declare r1, r2, c1, c2, variables and initialize other necessary variables.

Step 3: Input the no of rows and columns for the two 2-D array and store in r1, r2, c1, c2 respectively.

Step 4: If c1 is not equal to r1, then display “Cannot Perform Multiplication” and go to step 10 else go to step 5.

Step 5: Initialize two 2-D arrays to store the matrices to multiply.

Step 6: Initialize a 2-D array to store the product of the matrices.

Step 7: Input the elements of both the matrices

Step 8: Multiply the matrices using nested loops.

Step 9: Print the product in matrix form as console output.

Step 10: End

PROGRAM

```
 import java.util.Scanner;

class multiply

{

public static void main(String args[])

{

int r1, r2,c1,c2,i,j,k,sum;

Scanner in = new Scanner(System.in);


System.out.println("Enter the no. of Rows for First Matrix");

r1 = in.nextInt();


System.out.println("Enter the no. Columns for First Matrix");

c1 = in.nextInt();

System.out.println("Enter the no. of Rows for Second Matrix");

r2 = in.nextInt();


System.out.println("Enter the no. of Columns for Second Matrix");

c2 = in.nextInt();


if(c1==r2)

{


int mat1[][] = new int[r1][c1];

int mat2[][] = new int[r2][c2];

int res[][] = new int[r1][c2];
```

```
System.out.println("Enter the Elements for First Matrix");

for ( i= 0 ; i < r1 ; i++ )
{

for ( j= 0 ; j < c1 ;j++ )
mat1[i][j] = in.nextInt();

}
System.out.println("Enter the Elements for Second Matrix");

for ( i= 0 ; i < r2 ; i++ )
{

for ( j= 0 ; j < c2 ;j++ )
mat2[i][j] = in.nextInt();

}
in.close();

System.out.println("\nResult: ");
for ( i= 0 ; i < r1 ; i++ )

for ( j= 0 ; j <c2;j++)
{
```

```
sum=0;
for ( k= 0 ; k <r2;k++ )
{
sum +=mat1[i][k]*mat2[k][j] ;

}
res[i][j]=sum;
}
for ( i= 0 ; i < r1; i++ )
{
for ( j=0 ; j < c2;j++ )
System.out.print(res[i][j]+" ");

System.out.println();
}
}
else
System.out.print("Cannot Perform Multiplication");
}
}
```

RESULT

```
Enter the no. of Rows for First Matrix
2
Enter the no. Columns for First Matrix
2
Enter the no. of Rows for Second Matrix
```

2

Enter the no. of Columns for Second Matrix

2

Enter the Elements for First Matrix

1 2 3 4

Enter the Elements for Second Matrix

6 7 5 8

Result:

16 23

38 53

PROGRAM NO.6

AIM

To write a Java program to display the transpose of a given matrix

ALGORITHM:

Step 1: Start

Step 2: Declare row and column variables and initialize other necessary variables.

Step 3: Input the no of rows and columns for the two 2-D array and store in row and column respectively.

Step 4: Initialize a 2-D array to store the matrix.

Step 5: Input the elements of the matrix.

Step 6: Transpose the matrices using nested loops.

Step 7: Print the transpose of the matrix form as console output.

Step 8: End

PROGRAM

```
 import java.util.Scanner;
public class transpose
{
   public static void main(String args[])
   {
   int i, j;
   Scanner s = new Scanner(System.in);
   System.out.println("Enter the no. of Rows: ");
   int row = s.nextInt();
   System.out.println("Enter the no. of Columns: ");
   int column = s.nextInt();
   int array[][] = new int[row][column];
   System.out.println("Enter the Elements of the Matrix:");
   for(i = 0; i < row; i++)
   {
      for(j = 0; j < column; j++)
         { System.out.print(" ");
         array[i][j] = s.nextInt();


         }
   }
   System.out.println("Original Matrix: " );
   for(i = 0; i < row; i++)
      {
         for(j = 0; j < column; j++)
```

```
            {
            System.out.print(array[i][j]+" ");
            }
            System.out.println(" ");
        }
    System.out.println("Transpose Matrix: ");
    for(i = 0; i < column; i++)
        {
            for(j = 0; j < row; j++)
            {
                System.out.print(array[j][i]+" ");
            }
            System.out.println(" ");
        }
    }
}
```

RESULT

```
Enter the no. of Rows:
2
Enter the no. of Columns:
3
Enter the Elements of the Matrix:
 2 3 4 5 66 3
Original Matrix:
2 3 4
5 66 3
```

Transpose Matrix:

2 5

3 66

4 3

EXPERIMENT NO 7

AIM

Write a Java program to calculate the area of different shapes namely circle, rectangle, triangle using the concept of method overloading in class

ALGORITHM

Step 1: start

Step 2: define a class overloading with required members and methods

Step 3: create an overloaded method calcarea

Step 4: input the dimensions of circle , triangle and rectangleAIM

Step 5: find the area using menudriven function and using the respective formulas defined in overloaded methods

Step 6: print the result in output console

Step 7: stop

PROGRAM

```
import java.util.Scanner;

import java.lang.Math;

public class overloading {

double p;

double area;

   void calcarea(double r)

   {
```

```
    System.out.println("area of circle is : "+3.14*r*r );
   }
   void calcarea(int b, int h)
   {
    area=0.5*b*h;
    System.out.println("area of triangle is :"+ area);
   }
   void calcarea(float l, float m)
   {

    System.out.println("area of rectangle is : "+ l*m);
   }
   public static void main(String[] args) {
 int n;
      overloading ob=new overloading();
      Scanner s=new Scanner(System.in);
      do{
      System.out.println("1.area of circle");
      System.out.println("2.area of triangle");
      System.out.println("3.area of rectangle");
      System.out.println("4.exit");
System.out.println("enter your choice : ");
n=s.nextInt();
switch(n){
   case 1:
 System.out.println("enter radius : ");
```

```
double r=s.nextDouble();
ob.calcarea(r);
break;
case 2:
System.out.println(" enter height of triangle");
int h= s.nextInt();
System.out.println(" enter base length of triangle");
 int b= s.nextInt();
ob.calcarea(b,h);
break;
case 3:
 System.out.println(" enter side lenth and breadth of rectangle :");
float l= s.nextFloat();
float m= s.nextFloat();
ob.calcarea(l,m);
break;
case 4:
System.out.println(" exit ");
break;
}
 }while(n!=4);
    s.close();
   }
}
```

RESULT

1.area of circle

2.area of triangle

3.area of rectangle

4.exit

enter your choice :

1

enter radius :

3

area of circle is : 28.259999999999998

1.area of circle

2.area of triangle

3.area of rectangle

4.exit

enter your choice :

2

enter height of triangle

3

enter base length of triangle

4

area of triangle is :6.0

1.area of circle

2.area of triangle

3.area of rectangle

4.exit

enter your choice :

3

enter side lenth and breadth of rectangle :

4 5

area of rectangle is : 20.0

1.area of circle

2.area of triangle

3.area of rectangle

4.exit

enter your choice :

4

exit

Process finished.

EXPERIMENT-8

AIM

Write two Java classes Employee and Engineer. Engineer should inherit from Employeeclass. Employee class to have two methods display() and calcSalary(). Write a program to display the engineer salary and to display from Employee class using a single object in- stantiation (i.e., only one object creation is allowed).

● display() only prints the name of the class and does not return any value. Ex. " Name

of class is Employee."

● calcSalary() in Employee displays "Salary of employee is 10000" and calcSalary() in

Engineer displays "Salary of employee is 20000."

ALGORITHM:

Step 1: Start

Step 2: Define parent class Employee with display and calcsalary methods

Step 3: Define subclass engineer which extends from parent class Employee with same methods as in employee class

Step 4: Use super keyword in display and calcsalary method to call methods in parent class

Step 5: Object of subclass engineer is created in main method

Step 6: Call the methods using object of subclass

Step 7: End

PROGRAM:

```
class employee{
   void display(){
      System.out.println("Name of class is employee");
   }
   void calcsalary(){
      System.out.println("Salary of employee is 1000");
   }
}
class engineer extends employee{
  void display(){
    super.display();
      System.out.println("Name of class is engineer");
   }
   void calcsalary(){
super.calcsalary();
      System.out.println("Salary of engineer is 2000");
   }

}
```

```
public class inheritance {
    public static void main(String[] args) {
        engineer e=new engineer();
        e.display();
        e.calcsalary();

}
}
```

OUTPUT:

Name of class is employee

Name of class is engineer

Salary of employee is 1000

Salary of engineer is 2000

EXPERIMENT:9

AIM:

Write a Java program which creates a class named 'Employee' having the following members: Name, Age, Phone number, Address, Salary.

It also has a method named 'print- Salary( )' 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

ALGORITHM:

Step 1: Start

Step 2: Define parent class Employee with required members methods and construcor

Step 3: Initialize values to members using parametirized constructor

Step 3: Define subclass Officer which extends from class employee

Step 4: Add additional member specialization, constructor

Step 5: Initialize values to members using parametirized constructor of class officer and use super keyword to call constructor of employee class

Step 6: Define subclass manager which extends from class employee

Step 7: Add additional member department, constructor

Step 8: Initialize values to members using parametirized constructor of class manager and use super keyword to call constructor of employee class

Step 9: Accept values using scanner

Step 10:object of both classes officer and manager are created and the values passed to the resective constructors

Step 11: Displays the result by accessing the members of class and calling the printsal function of employee using the officer and manager class

Step 12: End

PROGRAM:

OUTPUT:

Enter the name of Officer

Rob

Enter the age of Officer

34

Enter the phone no: of Officer

612345768

Enter the address of Officer

Los Angeles,California,USA

Enter the salary of Officer

30000

Enter the specilization of Officer

Software engineer

Enter the name of Manager

Martin

Enter the age of Manager

47

Enter the phone no: of Manager

67340921

Enter the address of Manager

Brooklyn,New York,USA

Enter the salary of Manager

45000

Enter the department of Manager

Data Analytics

Details of oficer

salary 30000.0

Name:Rob

Address:Los Angeles,California,USA

Phone number:612345768

Age:34

Department:Software engineer

Details of manager

Salary: 45000.0

Name:Martin

Address:Brooklyn,New York,USA

Phone number:67340921

Age:47

Department:Data Analytics

EXPERIMENT NO 10

AIM

Write a program illustrating the working of command line arguments.
a.To find the sum of command line arguments and count the invalid integers entered.
b.To get the name using command line.

ALGORITHM

Step 1: Start
Step 2: define class commandsum
Step 3: Initialize s=0,inv=0
Step 4:Input the commandline argument
Step 5:start try block
Step 6: calculate sum s
Step 7:Catch block to catch numberformat exception
Step 8:calculate invalid integers
Step 9:print the result
Step 10:stop

PROGRAM

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

```
  {
        int s=0,inv=0;

        System.out.print("The commandline argument is ");
        for(int i=0;i<args.length;i++)
                System.out.print(args[i]);
        System.out.println();


        for(int i=0;i<args.length;i++)
        {
                try
                {
                        s+=Integer.parseInt(args[i]);
                }
                catch(NumberFormatException e)          {
                        inv++;
                }
        }
        System.out.println("The sum is "+s);
        System.out.println("The number of invalid integers is "+inv);
  }
}
```

RESULT

The commandline argument is 123a4
The sum is 10
The number of invalid integers is 1

EXPERIMENT NO 11

AIM

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 od sides in the given geometrical structures.

ALGORITHM

Step 1:Start
Step 2:Define an abstract class Shape where numberofsides method is declared
Step 3:Define the subclasses for Shape with definitions for the method declared in the parent class Shape
Step 4: Objects of subclass are declared in main method

Step 5: numberofsides method of the respective subclass is called using the respective objects
Step 6: End

PROGRAM

```
abstract class Shape
{
   abstract void numberOfSides();
}

class Rectangle extends Shape
{
        void numberOfSides()
        {
                System.out.println("The number of sides of Rectangle is 4");
        }
}

class Triangle extends Shape
{
   void numberOfSides()
   {
        System.out.println("The number of sides of Triangle is 3");
   }
}

class Hexagon  extends Shape
{
   void numberOfSides()
   {
        System.out.println("The number of sides of Hexagon is 6");
   }
}

public class Sides
{
   public static void main(String args[])
   {
        Rectangle r=new Rectangle();
        Triangle t= new Triangle();
        Hexagon h=new Hexagon();
        r.numberOfSides();
        t.numberOfSides();
        h.numberOfSides();

   }
}
```

RESULT

The number of sides of Rectangle is 4
The number of sides of Triangle is 3
The number of sides of Hexagon is 6

EXPERIMENT NO 12

AIM

Write a java program to demonstrate the use of garbage collector

ALGORITHM

Step 1: Start
Step 2:Define a class GarbageCollector with method finalize to find the garbage collected
Step 3: 2 Objects of garbagecollector class is declared
Step 4: Each of the objects is assigned a NULL value
Step 5: End

PROGRAM

```
public class GarbageCollector
{
   public void finalize()
   {
         System.out.println("Garbage collected "+this);
   }
   public static void main(String args[])
   {
         GarbageCollector g1=new GarbageCollector();
         GarbageCollector g2=new GarbageCollector();
         g1=null;
         g2=null;
         System.gc();
   }
}
```

RESULT

Garbage collected GarbageCollector@2130772
Garbage collected GarbageCollector@cd4e940

Experiment no 13

AIM

Write a class that implements the CharSequence interface found in the java.lang package. Your implementation should return the string backwards.

charAt(int index)

Returns the character at the specified index.

Int length()

Returns the length of this character sequence.

CharSequence subSequence(int start, int end)

Returns a new character sequence that is a subsequence of this sequence.

String toString()

Returns a string containing the characters in this sequence in the same order as this sequence.

ALGORITHM

Step 1: Start

Step 2: create new object -> s= new object()

Step 3: Read the string

Step 4: for i=s.length()-1 to 0

Step 5: print s.charAt(i)

Step 6: Loop

Step 7: print using subsequence method

Step 8: print using toString method

Step 9: End

PROGRAM

```
import java.util.*;

public class langinterface implements CharSequence {
```

```
private String s;

public langinterface() {
   Scanner sc = new Scanner(System.in);
   System.out.println("Enter the string:");
   s = sc.nextLine();
}

public char charAt(int i) {
   return s.charAt(i);
}

public int length() {
   return s.length();
}

public CharSequence subSequence(int start, int end) {

   StringBuilder sub = new StringBuilder(s.subSequence(start, end));
   return sub.reverse();
}

public String toString() {
   StringBuilder s = new StringBuilder(this.s);
   return s.reverse().toString();
}
```

```
    public static void main(String[] args) {
        langinterface s = new langinterface();

        for (int i = s.length() - 1; i >= 0; i--) {
            System.out.print(s.charAt(i));
        }
        System.out.println();

        int start = 0;
        int end = s.length();
        System.out.println(s.subSequence(start, end));

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

    }
}
```

OUTPUT

```
Enter the string:
palindrome
emordnilap
emordnilap
emordnilap
```

Experiment 14

AIM

Write a java program that shows how to create a user-defined exception using try, catch, throw and finally.

ALGORITHM

Step 1: Start

Step 2: define class Test that has constructor and method toString() and extends Exception class

Step 3: On the main class start try statement

Step 4: throw new Test obj

Step 5: start catch block

Step 6: print the exception that is caught

Step 7: start finally block

Step 8: End

PROGRAM

```
class Test extends Exception{
   String a;
   Test(String b){
      a=b;
   }
   public String toString(){
      return("Exception occured:"+a);
   }
}
public class Tryc {
   public static void main(String[] args){
```

```
        try{
            System.out.println("Try block started");
            throw new Test("TEST EXCEPTION HERE");
        }
        catch(Test exp){
            System.out.println("Catch Block STARTED!");
            System.out.println(exp);
        }
        finally{
          System.out.println("THIS IS FINALLY BLOCK");
        }

    }

}
```

OUTPUT

```
Try block started
Catch Block STARTED!
Exception occured:TEST EXCEPTION HERE
THIS IS FINALLY BLOCK
```

EXPERIMENT NO 15

AIM

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

ALGORITHM

Step 1: Start
Step 2: Start try block

Step 3: Create new file
Step 4: Catch block to catch IO exception
Step 5: Start try block
Step 7: Enter text to be entered
Step 8: Write in to the File
Step 9: Catch block to catch IO exception
Step 10: Start try block
Step 11: Close the file
Step 12: Catch block to catch IO exception
Step 13: Start try block
Step 14: open file in new file object
Step 15: Catch block to catch IO exception
Step 16: Start try block
Step 17: Print the contents of the file
Step 18: Catch block to catch IO exception
Step 19: Start try block
Step 20: Close the file
Step 21: Catch block to catch IO exception
Step 22:Stop

PROGRAM

```
import java.io.*;
import java.util.Scanner;
public class file {
   public static void main(String[] args)
   {
          int i;
          Scanner sc =new Scanner(System.in);
          String str;
          FileInputStream fin=null;
          FileOutputStream fout=null;

          try
          {
                  File myObj=new File("file1.txt");
                  if(myObj.createNewFile())
                          System.out.println("Created file "+myObj.getName());
          }
          catch(IOException e)
          {
                  System.out.println("Error creating file");
          }

          try
          {
                  System.out.print("Enter the text ");
                  str=sc.nextLine();
```

```
        byte s[]=str.getBytes();
        fout=new FileOutputStream("file1.txt");
        fout.write(s);
}
catch(IOException e)
{
        System.out.println("Error writing file");
}

try
{
        if(fout!=null)
                fout.close();
}
catch(IOException e)
{
        System.out.println("Error closing file");
}

try
{
fin=new FileInputStream("file1.txt");
}
catch(FileNotFoundException e)
{
        System.out.println("Error opening file");
}

try
{
        do {
                i=fin.read();
                if(i==-1)
                {
                        System.out.println();
                        System.out.println("End of file");
                }
                else
                        System.out.print((char)i);
        }while(i!=-1);
}
catch(IOException e)
{
        System.out.println("Error reading file");
}

try
```

```
        {
                if(fin!=null)
                        fin.close();
        }
        catch(IOException e)
        {
                System.out.println("Error closing file");
        }
   }
}
```

RESULT

```
Enter the text JAVA PROGRAM ON FILES
JAVA PROGRAM ON FILES
End of file
```

EXPERIMENT NO 16

AIM

Write a java program that reads a line of integers,and then displays each integer,and the sum of all the integers(Use String Tokenizer class of java.util).

ALGORITHM

Step 1: Start
Step 2: define class strtoken
Step 3: Initialize sum=0
Step 4:Input the string and token through scanner
Step 5:find the sum using while loop
Step 9: Print the sum in output console
Step 10: End

PROGRAM

```
import java.util.Scanner;
import java.util.StringTokenizer;
public class Strtoken
{
   public static void main(String args[])
   {
        int sum=0,x;
        Scanner sc=new Scanner(System.in);
```

```
        System.out.print("Enter the string ");
        String s=sc.nextLine();
        System.out.print("Enter the token ");
        String t=sc.nextLine();

        StringTokenizer c=new StringTokenizer(s,t);

        while(c.hasMoreTokens())
        {
                x=Integer.parseInt(c.nextToken());
                System.out.println(x);
                sum+=x;
        }
        System.out.println("The sum is "+sum);
   }
}
```

RESULT

```
Enter the string 1+2+3+4+5
Enter the token +
1
2
3
4
5
The sum is 15
```

EXPERIMENT NO 17

AIM

Write a Java program that reads a file and displays the file on the screen, with a line number before each line.

ALGORITHM

Step 1: Start

Step 2: read the file

Step 3: Start for loop

Step 4: Read line by line using Read()

Step 5: print the line numbered

Step 6: Close the file

Step 7: End

PROGRAM

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


public class lab17 {
   public static void main(String[] args) throws IOException {
      int j = 1;
      char ch;
      Scanner scan = new Scanner(System.in);
      System.out.print("\nEnter File name: ");
      String str = scan.next();
      FileInputStream f = new FileInputStream(str);
      System.out.println("\nContents of the file are");
      int n = f.available();
      System.out.print("line "+j + " contains: ");
      for (int i = 0; i < n; i++) {
         ch = (char) f.read();
         System.out.print(ch);
         if (ch == '\n') {
            System.out.print("line "+ ++j + " contains: ");


         }


      }
      f.close();
      scan.close();


   }
```

}

RESULT

Enter the file name: text.txt

Contents of the file are:

line 1 contains: Hello world.

line 2 contains: Hi java.

line 3 contains: Goodbye 2020.

EXPERIMENT NO 18

AIM

Write a Java program that displays the number of characters, lines and words in a text file.

ALGORITHM

step1:start

step 2: Open the file

step 3: Read the file char by char

step 4: if char == '\n' new line

Step 5: if char == ' ' new word

Step 6: else no_of_characters ++

Step 7: close the file

Step 8:End

PROGRAM

```
package filestring;
import java.io.*;
public class filestring {

        public static void main(String[] args) {
                int i,ch=0,lin=1,word=1;
```

```
FileInputStream fin;
try {
        fin=new FileInputStream(args[0]);
}
catch(FileNotFoundException e) {
        System.out.println("File not found");
        return;
}
try
{
        System.out.println("Contents of file is");
         do
        {
        i=fin.read();
        if(i!=-1)
        {
                System.out.print((char)i);
                switch((char)i)
                {
                case '\n':lin++;
                                                word++;
                                                break;
                case ' ':word++;
                                                break;

                default:ch++;
                }
        }
        }while(i!=-1);
}
```

```
                catch(IOException e)
                {
                        System.out.println("Error in reading file");
                }
                System.out.println();
                System.out.println();
                System.out.println("Number of characters : "+ (ch-lin+1));
                System.out.println("Number of word : "+word);
                System.out.println("Number of lines :"+lin);
                try
                {
                        fin.close();
                }
                catch(IOException e)
                {
                        System.out.println("Error in closing file");
                }
        }

}
```

OUTPUT

EXPERIMENT NO 19

AIM

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

Step 1: Start

Step 2: Create a list

Step 3: accept choice from the user

Step 4: Add and delete from front and back according to input choice

Step 5: display after each changes

Step 6:Exit when user inputs exit

Step 7: end

PROGRAM

```
 import java.util.*;
import java.lang.*;
/**
 * J19
 */
public class J19 {


    public static void main(String[] args) {
        LinkedList<Integer>list = new LinkedList<Integer>();
        Scanner sc =new Scanner(System.in);
        int ch,n;
        while (true) {
            System.out.println();
            System.out.println("DOUBLY LINKED LIST");
            System.out.println("1. Insert in Front");
            System.out.println("2. Insert in end");
            System.out.println("3. Delete from front");
            System.out.println("4. Delete from last");
            System.out.println("5. Display");
            System.out.println("6. Exit out");
            System.out.println("Enter your choice...");
            ch=sc.nextInt();
```

```
    switch (ch) {
        case 1:
            System.out.println("Enter the element to insert:");
            n=sc.nextInt();
            list.addFirst(n);
            break;
        case 2:
            System.out.println("Enter the element to insert:");
            n=sc.nextInt();
            list.addLast(n);
            break;
        case 3:
            list.removeFirst();
            break;
        case 4:
            list.removeLast();
            break;
        case 5:
            System.out.println("Doubly Linked List:");
            for(Integer x: list)
                System.out.print(x+" ");
            System.out.println();
             break;
        case 6:
            sc.close();
             System.exit(0);
        default:
            break;
    }
}
```

```
    }
}
```

RESULT

it is a

EXPERIMENT NO 20

AIM

Write a Java program that implements the Quicksort algorithm for sorting a list of names in ascending order.

ALGORITHM

Step 1: Start

Step 2: Read the names

Step 3: Create partitions for quicksort

Step 4: group elements < pivot to left and others to right

Step 5: recall same function recursively on the splitted 2 lists

Step 6: repeat step 4 and 5 till partitions are done

Step 7: Remerge the list

Step 8: Print the list

Step 9: End

PROGRAM

```
import java.util.Scanner;


public class lab20 {
    int partition(String arr[], int low, int high) {
        String pivot = arr[high];
        int i = low - 1;
        for (int j = low; j < high; j++) {
            if (arr[j].compareToIgnoreCase(pivot) < 0) {
```

```
            i++;
            String temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
    }
    String temp = arr[i + 1];
    arr[i + 1] = arr[high];
    arr[high] = temp;

    return i + 1;
}

void quickSort(String arr[], int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    String arr[] = new String[20];

    System.out.println("Enter the number of names");
    int no = sc.nextInt();
```

```
        sc.nextLine();

        System.out.println("\nEnter the names");
        for (int i = 0; i < no; i++)
            arr[i] = sc.nextLine();

        lab20 qs = new lab20();
        qs.quickSort(arr, 0, no - 1);

        System.out.println("\nSorted list of names:");
        for (int i = 0; i < no; i++)
            System.out.println(arr[i]);
    }
}
```

RESULT

```
Enter the number of names: 3
Enter the names
Raju
Bheem
Jaggu
Sorted list of names:
Bheem
Jaggu
Raju
```

EXPERIMENT NO 21

AIM

Write a Java program that reads a file and displays the file on the screen, with a line number before each line.

ALGORITHM

Step 1: Start

Step 2:

Step 3:

PROGRAM

RESULT

it is a

PROGRAM 22

AIM

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, the second thread computes the square of the number and prints. If the value is odd the third thread will print the value of the cube of the number.Use thread synchronization.

ALGORITHM

step 1:start

step 2: Start thread 1

step 3: n=Generate random number

Step 4:if n is odd run thread 2 to find square

Step 5: else n is even run thread 3 to find cube

Step 6: End

PROGRAM

```
package multithread.java;
import java.util.Random;
class Number extends Thread
{
 public synchronized void run()
 {
 Random random = new Random();
 for(int i =0; i<10; i++)
 {
 int randomInteger = random.nextInt(100);
 System.out.println("\nRandom Integer generated : " + randomInteger);
 if ((randomInteger%2)==0)
 {
 Square s = new Square(randomInteger);
 s.start();
 }
 else
 {
 Cube c = new Cube(randomInteger);
 c.start();
 }
 try {
 Thread.sleep(1000);

 } catch (InterruptedException ex) {
 System.out.println(ex);
}
 }
 }
```

```
}
class Square extends Thread
{
 int x;
Square(int n)
 {
 x = n;
 }
 public synchronized void run()
 {
 int sqr = x * x;
 System.out.println("Square of " + x + " : " + sqr );
 }
}
class Cube extends Thread
{
 int x;
 Cube(int n)
 {
 x = n;
 }
 public synchronized void run()
 {
 int cub = x * x * x;
 System.out.println("Cube of " + x + " : " + cub );
 }
}
public class multithread {
 public static void main(String args[])
```

```
 {
 Number n = new Number();
 n.start();
 }
}
```

OUTPUT

PROGRAM 23

AIM:

Write a Java program to create two threads: One for displaying all odd numbers between 1and 100 and second thread for displaying all even numbers between 1 and 100, Use thread priorities in the program.

ALGORITHM

step1: Start

step 2: select start thread 1 by shared printer class

step 3: print i from for i=1 ; i<=100 ;i+=2

step 4: Start thread 2

Step 5 print i from for i=2 ; i<=100; i+=2

Step 6: Stop

PROGRAM

```
package oddeventhread;
class OddThread extends Thread
{
int limit;
sharedPrinter printer;
public OddThread(int limit, sharedPrinter printer)
{
```

```
this.limit = limit;
this.printer = printer;
}
@Override
public void run()
{
int oddNumber = 1;
while (oddNumber <= limit)
{
printer.printOdd(oddNumber);
oddNumber = oddNumber + 2;
}
}
}
class EvenThread extends Thread
{
int limit;
sharedPrinter printer;
public EvenThread(int limit, sharedPrinter printer)
{
this.limit = limit;
this.printer = printer;
}
@Override
public void run()
{
int evenNumber = 2;
while (evenNumber <= limit)
{
```

```
printer.printEven(evenNumber);
evenNumber = evenNumber + 2;
}
}
}
class sharedPrinter
{
boolean isOddPrinted = false;
synchronized void printOdd(int number)
{
while (isOddPrinted)
{
try
{
wait();
}
catch (InterruptedException e)
{
e.printStackTrace();
}
}
System.out.println(Thread.currentThread().getName()+" : "+number);
isOddPrinted = true;
try
{
Thread.sleep(1000);
}
catch (InterruptedException e)
{
```

```
e.printStackTrace();
}
notify();
}
synchronized void printEven(int number)
{
while (! isOddPrinted)
{
try
{
wait();
}
catch (InterruptedException e)
{
e.printStackTrace();
}
}
System.out.println(Thread.currentThread().getName()+" : "+number);
isOddPrinted = false;
try
{
Thread.sleep(1000);
}
catch (InterruptedException e)
{
e.printStackTrace();
}
notify();
}
```

```
}
public class oddeventhread
{
public static void main(String[] args)
{
sharedPrinter printer = new sharedPrinter();
OddThread oddThread = new OddThread(100, printer);
oddThread.setName("Odd-Thread");
EvenThread evenThread = new EvenThread(100, printer);
evenThread.setName("Even-Thread");
oddThread.start();
evenThread.start();
}
}
```

OUTPUT

PROGRAM 24

AIM

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. Use Java Swing.

ALGORITHM

Step 1: Create the buttons 0-9 and +,-,.,*,/,C

Step 2:add textfiled to layout

Step 3: add the created buttons

Step 4: Accept input

Step 5: process input and display result

Step 6: clear result when C is pressed

Step 7:End

PROGRAM

```
package calculator.java;
import java.awt.event.*;
import javax.swing.*;
import java.awt.*;
class calculator extends JFrame implements ActionListener {
	static JFrame f;
	static JTextField l;
	String s0, s1, s2;
	calculator()
	{
		s0 = s1 = s2 = "";
	}
	public static void main(String args[])
	{
		f = new JFrame("calculator");

		try {

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
		}
		catch (Exception e) {
			System.err.println(e.getMessage());
```

```
        }
        calculator c = new calculator();
        l = new JTextField(16);
        l.setEditable(false);
        JButton b0, b1, b2, b3, b4, b5, b6, b7, b8, b9, ba, bs, bd, bm, be, beq,
beq1;
        b0 = new JButton("0");
        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");
        beq1 = new JButton("=");
        ba = new JButton("+");
        bs = new JButton("-");
        bd = new JButton("/");
        bm = new JButton("*");
        beq = new JButton("C");
        be = new JButton(".");
        JPanel p = new JPanel();
        bm.addActionListener(c);
        bd.addActionListener(c);
        bs.addActionListener(c);
```

```
ba.addActionListener(c);
b9.addActionListener(c);
b8.addActionListener(c);
b7.addActionListener(c);
b6.addActionListener(c);
b5.addActionListener(c);
b4.addActionListener(c);
b3.addActionListener(c);
b2.addActionListener(c);
b1.addActionListener(c);
b0.addActionListener(c);
be.addActionListener(c);
beq.addActionListener(c);
beq1.addActionListener(c);
p.add(l);
p.add(ba);
p.add(b1);
p.add(b2);
p.add(b3);
p.add(bs);
p.add(b4);
p.add(b5);
p.add(b6);
p.add(bm);
p.add(b7);
p.add(b8);
```

```
        p.add(b9);
        p.add(bd);
        p.add(be);
        p.add(b0);
        p.add(beq);
        p.add(beq1);
        p.setBackground(Color.yellow);
        f.add(p);

        f.setSize(200, 220);
        f.show();
}
public void actionPerformed(ActionEvent e)
{
        String s = e.getActionCommand();
        if ((s.charAt(0) >= '0' && s.charAt(0) <= '9') || s.charAt(0) == '.') {
                if (!s1.equals(""))
                        s2 = s2 + s;
                else
                        s0 = s0 + s;
                l.setText(s0 + s1 + s2);
        }
        else if (s.charAt(0) == 'C') {
        s0 = s1 = s2 = "";
                l.setText(s0 + s1 + s2);
        }
```

```
            else if (s.charAt(0) == '=') {

                double te;
                if (s1.equals("+"))
                    te = (Double.parseDouble(s0) + Double.parseDouble(s2));
                else if (s1.equals("-"))
                    te = (Double.parseDouble(s0) - Double.parseDouble(s2));
                else if (s1.equals("/"))
                    te = (Double.parseDouble(s0) / Double.parseDouble(s2));
                else
                    te = (Double.parseDouble(s0) * Double.parseDouble(s2));
                l.setText(s0 + s1 + s2 + "=" + te);
                s0 = Double.toString(te);

                s1 = s2 = "";
            }
            else {
                if (s1.equals("") || s2.equals(""))
                    s1 = s;
            else {
                    double te;
                    if (s1.equals("+"))
                        te = (Double.parseDouble(s0) +
Double.parseDouble(s2));
                    else if (s1.equals("-"))
                        te = (Double.parseDouble(s0) -
Double.parseDouble(s2));
```

```
                                else if (s1.equals("/"))
                                        te = (Double.parseDouble(s0) /
Double.parseDouble(s2));
                                else
                                        te = (Double.parseDouble(s0) *
Double.parseDouble(s2));
                                s0 = Double.toString(te);


                            s1 = s;
                                s2 = "";
                        }
        l.setText(s0 + s1 + s2);
                    }
         }
}
```

OUTPUT

PROGRAM 25

AIM

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

Step 1: Create light lay out using signal

Step 2: create 3 radio buttons corresponding to each colors

Step 4: Add the signal and buttons to the jframe

Step 5: turn signal on and out acoording to Radio button

Step 6: end

PROGRAM

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
public class TrafficLight extends JFrame implements ActionListener {
   JRadioButton buttonRed, buttonorange, buttonGreen;
   Signal green = new Signal(Color.green);
   Signal orange = new Signal(Color.orange);
   Signal red = new Signal(Color.red);
   public TrafficLight(){
      setLayout(new FlowLayout());
      buttonRed = new JRadioButton("Red");
      buttonorange = new JRadioButton("orange");
      buttonGreen = new JRadioButton("Green");
      buttonRed.addActionListener(this);
      buttonorange.addActionListener(this);
      buttonGreen.addActionListener(this);
      JPanel trafficPanel = new JPanel(new GridLayout(3,1));
      trafficPanel.add(red);
      trafficPanel.add(orange);
      trafficPanel.add(green);
      JPanel lightPanel = new JPanel(new FlowLayout());
      lightPanel.add(buttonRed);
      lightPanel.add(buttonorange);
      lightPanel.add(buttonGreen);
```

```
    add(trafficPanel);
    add(lightPanel);
    pack();
    setVisible(true);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
public static void main(String[] args){
    new TrafficLight();
}
public void actionPerformed(ActionEvent e){
    if (e.getSource() == buttonRed){
        buttonGreen.setSelected(false);
        buttonorange.setSelected(false);
        green.turnOn(false);
        orange.turnOn(false);
        red.turnOn(true);
    } else if (e.getSource() == buttonorange){
        buttonGreen.setSelected(false);
        buttonRed.setSelected(false);
        orange.turnOn(true);
        green.turnOn(false);
        red.turnOn(false);
    } else if (e.getSource() == buttonGreen){
        buttonRed.setSelected(false);
        buttonorange.setSelected(false);
        red.turnOn(false);
```

```
            orange.turnOn(false);
            green.turnOn(true);
        }
    }
}
class Signal extends JPanel{

    Color on;
    boolean change;

    Signal(Color color){
        on = color;
        change = false;
    }

    public void turnOn(boolean a){
        change = a;
        repaint();
    }
    public Dimension getPreferredSize(){
        int size = (50)*2;
        return new Dimension( size, size );
    }
    public void paintComponent(Graphics g){

        g.setColor(Color.black);
```

```
        g.fillRect(0,0,150,250);
        if (change){
           g.setColor( on );
        } else {
           g.setColor(Color.white);
        }
        g.fillOval(10, 10, 80, 80);
        g.setColor(Color.blue);
        g.drawOval(10,10,80,80);
    }
}
```

OUTPUT

PROGRAM 26

AIM

Write a program to accept rollno, name, CGPA of “n” students and store the data to a database using JDBC connectivity. Display the list of students having CGPA greater than 7. (Use MySQL /PostgreSQL)

ALGORITHM

Step 1: connect with java data base using Connection

Step 2: Execute the Query

Step 3: print the result as string

Step 4: end

PROGRAM

```
package mysql.java;
import java.sql.*;
import java.util.*;
class mysql{
public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
try{
        String name="";
        int rollno;
        double cgpa;
Class.forName("com.mysql.jdbc.Driver");
Connection con=DriverManager.getConnection(
"jdbc:mysql://localhost:3306/jlab","root","root");
System.out.println("Enter the number of Students");
int n=sc.nextInt();
Statement stmt=con.createStatement();
for(int i=0;i<n;i++) {
        sc.nextLine();
        System.out.println("Enter the name::");
        name=sc.nextLine();
        System.out.println("Enter the rollno::");
        rollno=sc.nextInt();
        System.out.println("Enter the cgpa::");
        cgpa=sc.nextDouble();
String sql = "insert into studentdetails " +
      "VALUES ("+rollno+","'+name+"',"+cgpa+");";
```

```
stmt.executeUpdate(sql);}
ResultSet rs=stmt.executeQuery("select * from studentdetails where cgpa>7;");
while(rs.next())
System.out.println(rs.getInt(1)+"  "+rs.getString(2)+" "+rs.getDouble(3));
con.close();
}catch(Exception e){ System.out.println(e);}
}
}
```

OUTPUT