EXPT. NO. 1	**TO PRINT HELLOWORLD**	DATE: 11-09-2020
Filename :Hello.java	Compiler:javac/gdb	

AIM

To print helloworld .

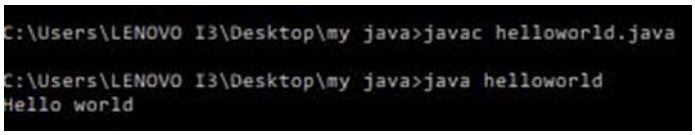
ALGORITHM

Step 1:Start
Step 2: Declare a class with name helloworld.
Step 3:Declare the main method.
Step 4:Display the string “Helloworld” using the method System.out.println.
Step 5:Stop.

PROGRAM

```
class helloworld
{
public static void main(String[] args)
{
System.out.println("Hello world");
}
}
```

OUTPUT



RESULT

The program was successfully executed and familiarized method to print string in java.

EXPT. NO.2	**TO PRINT AREA OF TRIANGLE**	DATE: 11-09-2020
Filename : area.java	Compiler:javac/gdb	

AIM

To print the area of triangle.

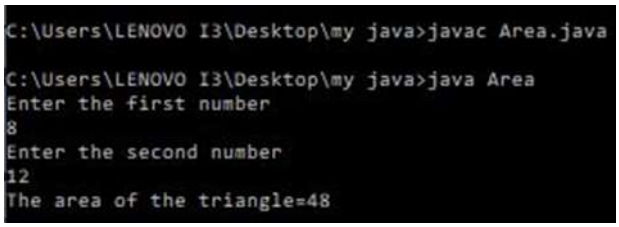
ALGORITHM

Step 1:Start
Step 2: Declare the class name Area.
Step 3: Declare the main method public static void main(String[] args).
Step 4: Read the base and height of the triangle.
Step 5: Calculate the area using A=bh/2.
Step 6: Print the result.

PROGRAM

```
import java.util.Scanner;
public class Area
{
public static void main(String[] args)
{
float b,h,a;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the first number");
b=sc.nextInt();
System.out.println("Enter the second number");
h=sc.nextInt();
a=(b*h)/2;
System.out.println("The area of the triangle="+a);
}
}
```

OUTPUT



RESULT

The program was successfully executed and familiarized method to print string in java.

EXPT. NO.3	**TO FIND THE LARGEST OF THREE NUMBERS**	DATE: 11-09-2020
Filename : largest.java	Compiler:javac/gdb	

AIM

To find the largest of three numbers.

ALGORITHM

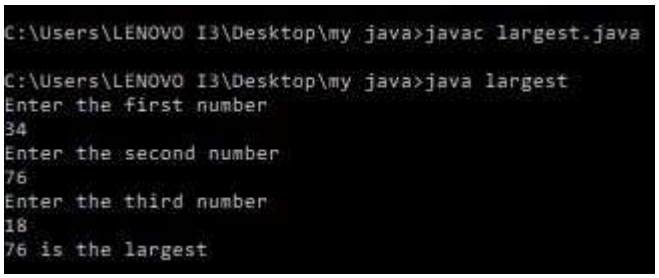
Step 1:Start
Step 2: Declare the class name Area.
Step 3: Declare the main method public static void main(String[] args).
Step 4: Read the first, second and third numbers
Step 5: Compare the numbers
Step 6: If the first number is greater than second and third, print first number.
Step 7: If the second number is greater than first and third, print second number.
Step 8: Else print the third number.

PROGRAM

```
import java.util.Scanner;
public class largest
{
public static void main(String[] args)
{
int a,b,c;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the first number");
a=sc.nextInt();
System.out.println("Enter the second number");
b=sc.nextInt();
System.out.println("Enter the third number");
c=sc.nextInt();
if(a>b&&a>c)
System.out.println(a+" is the largest");
else if(b>a&&b>c)
```

```
System.out.println(b+" is the largest");
else
System.out.println(c+" is the largest");
}
}
```

OUTPUT



RESULT

The program was successfully executed and familiarized method to print string in java.

EXPT. NO. 4	**PALINDROME STRING OR NOT**	DATE: 19-09-2020
Filename : pallindrome.java	Compiler:javac	

AIM

To check whether the given string is palindrome or not.

ALGORITHM

Step 1:Start
Step 2: Declare a class with name palindrome.
Step 3:Declare the main method.
Step 4:Read the string.
Step 5:Find the length of the string.
Step 6:Assign i as i-1.
Step 7:Thus store the reverse of the string in r, by updating the value of i.
Step 8:If the reverse of the string is equal to the input stirng then
Print “The string is palindrome”
Step 9:Otherwise
Print”The string is not palindrome”.
Step 10:Stop.

PROGRAM

```
import java.util.Scanner;
public class pallindrome
{
public static void main(String[] args)
{
String a,b="";
Scanner sc=new Scanner(System.in);
System.out.println("Enter the string");
a=sc.nextLine();
int x=a.length();
for(int i=x-1;i>=0;i--)
{
b=b+a.charAt(i);
}
```

```
if(a.equalsIgnoreCase(b))
{
System.out.println("The string is pallindrome");
}
else
{
System.out.println("The string is not pallindrome");
}
}
}
```

OUTPUT

RESULT

The program was successfully executed and familiarized method to print string in java.

EXPT. NO.5	**TO FIND THE FREQUENCY OF GIVEN CHARACTER**	DATE: 19-09-2020
Filename : frequency.java	Compiler:javac	

AIM

To find the frequency of given character.

ALGORITHM

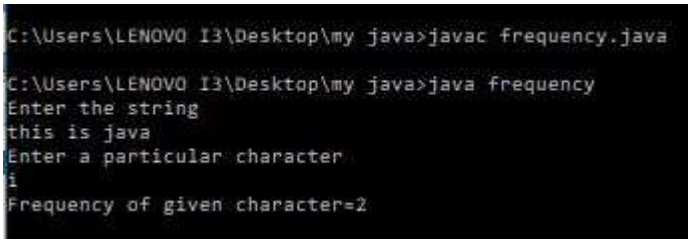
Step 1:Start
Step 2:Declare a class with name frequency
Step 3:Declare the main method.
Step 4:Read the string.
Step 5:Enter the character from the user whose frequency in the string is to be found and read it.
Step 6:Declare a variable count to store the frequency of the character in the string and assign it as 0.
Step 7:Calculate the length of the string and assign it as int i.
Step 8:Assign int i as 0.
Step 9:Compare the input character with each character in the string by updating the value of i.
Step 10:If a match is found,increment the value of count by 1.
Step 11:Repeat the process till i is less than 1.
Step 12:Print the value of count.
Step 13:Stop.

PROGRAM

```
import java.util.Scanner;
public class frequency
{
public static void main(String[] args)
{
System.out.println("Enter the string");
Scanner sc=new Scanner(System.in);
String str=sc.nextLine();
System.out.println("Enter a particular character");
char ch=sc.nextLine().charAt(0);
int count=0;
for(int i=0;i<str.length();i++)
{
```

```
if(ch==str.charAt(i))
{
count++;
}
}
System.out.println("Frequency of given character="+count);
}
}
```

OUTPUT



RESULT

The program was successfully executed and familiarized method to print string in java.

EXPT. NO.6	**TO MULTIPLY TWO GIVEN MATRIX**	DATE: 19-09-2020
Filename : matrixmul.java	Compiler:javac	

AIM

To multiply the given two matrices.

ALGORITHM

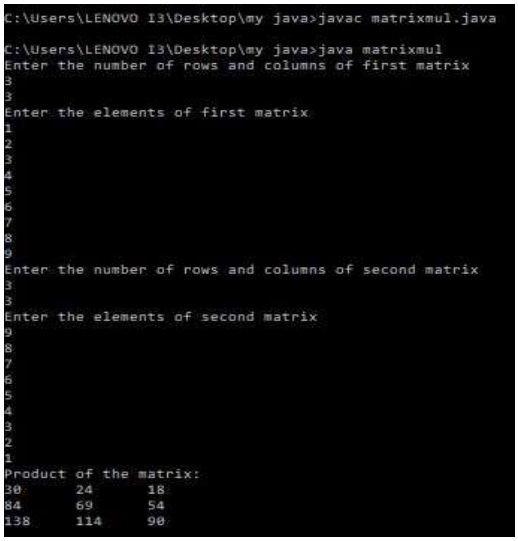
Step 1:Start
Step 2:Declare a class with name matrixmul.
Step 3:Declare the main method.
Step 4:Declare two variable i and j for entering the elements into rows and columns.
Step 5:Declare a matrix c to store the resultant.
Step 6:Enter the rows and columns of first matrix as r1 and c1.
Step 7:Enter the elements of first matrix.
Step 8: Enter the rows and columns of second matrix as r2 and c3.
Step 9: Enter the elements of second matrix.
Step 10:If r1=c2 then multiply the two matricesand store the resultant in matrix c.
Print the matrix c
Step 11:else
Print “Multiplication is not possible
Step 12:Stop.

PROGRAM

```
import java.util.Scanner;
import java.lang.*;
public class matrixmul
{
public static void main(String[] args)
{
Scanner sc=new Scanner(System.in);
int m,n,p,q,sum=0,c,d,k;
System.out.println("Enter the number of rows and columns of first matrix");
m=sc.nextInt();
n=sc.nextInt();
int first[][]=new int[m][n];
System.out.println("Enter the elements of first matrix");
```

```
for(c=0;c<m;c++)
for(d=0;d<n;d++)
first[c][d]=sc.nextInt();
System.out.println("Enter the number of rows and columns of second matrix");
p=sc.nextInt();
q=sc.nextInt();
if(n!=p)
System.out.println("The matrices can't be multiplied");
else
{
int second[][]=new int[p][q];
int multiply[][]=new int[m][n];
System.out.println("Enter the elements of second matrix");
for(c=0;c<p;c++)
for(d=0;d<q;d++)
second[c][d]=sc.nextInt();
for(c=0;c<m;c++)
{
for(d=0;d<q;d++)
{
for(k=0;k<p;k++)
sum=sum+first[c][k]*second[k][d];
multiply[c][d]=sum;
sum=0;
}
}
System.out.println("Product of the matrix:");
for(c=0;c<m;c++)
{
for(d=0;d<q;d++)
System.out.print(multiply[c][d]+"\t");
System.out.print("\n");
}
}
}
}
```

OUTPUT



RESULT

The program was successfully executed and familiarized method to print string in java.

EXPT. NO.7	**TO CREATE CLASS EMPLOYEE THAT INHERITS OFFICER AND MANAGER**	DATE: 26-09-2020
Filename : office.java	Compiler:javac/gdb	

AIM

To create a class employee that inherits class officer and class manager.

ALGORITHM

ALGORITHM FOR CLASS employee

Step 1: Create a class employee with instance variables name,age,phno,address and salary
Step 2: Initialize the instance variables using parameterized constructor.
Step 3: Define method display to print the values of data members.
Step 4: Define method printsalary to display value of data member salary.
Step 5: Create class officer and manager that inherits Employee class.
Step 6: Initialize data members specialization and department of officer clas and manager using constructors
Step 7: Invoke display method of parent class from both child classes.

ALGORITHM FOR CLASS office

Step 1: Declare the main method.
Step 2: Inside main, create objects of both child classes officer and manager with the required attributes, that are set in the constructor.
Step 3: Invoke necessary methods of officer class to display the values of instance variables.

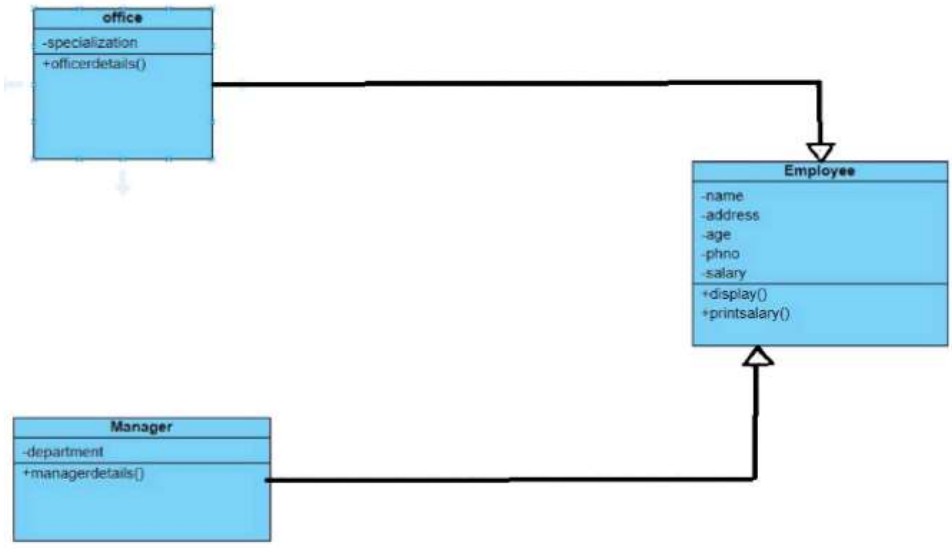
PROGRAM

```
import java.util.Scanner;
class Employee
{
String name;
String address;
int age;
```

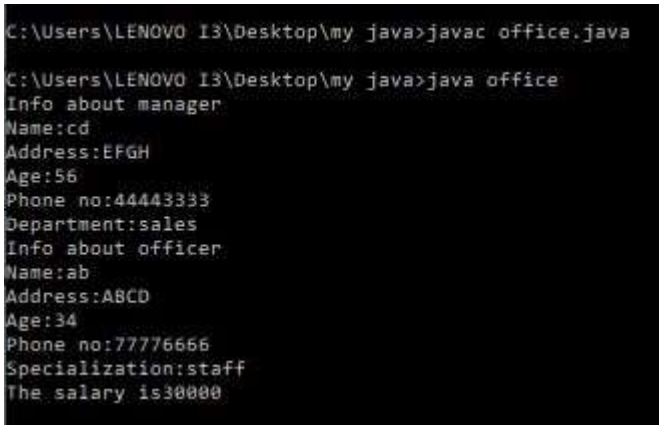
```
int phno;
int salary;
Employee(String n,String ad,int ag,int ph,int sal)
{
name=n;
address=ad;
age=ag;
phno=ph;
salary=sal;
}
public void display()
{
System.out.println("Name:"+name);
System.out.println("Address:"+address);
System.out.println("Age:"+age);
System.out.println("Phone no:"+phno);
}
public void printsalary()
{
System.out.println("The salary is"+salary);
}
}
class officer extends Employee
{
String specialization;
officer(String n,String ad,int ag,int ph,int sal,String spec)
{
super(n,ad,ag,ph,sal);
specialization=spec;
}
void officerdetails()
{
display();
System.out.println("Specialization:"+specialization);
}
}
class manager extends Employee
{
String department;
manager(String n,String ad,int ag,int ph,int sal,String dep)
{
super(n,ad,ag,ph,sal);
department=dep;
```

```
}
void managerdetails()
{
display();
System.out.println("Department:"+department);
}
}
class office
{
public static void main(String args[])
{
officer o=new officer("ab","ABCD",34,77776666,30000,"staff");
manager m=new manager("cd","EFGH",56,44443333,60000,"sales");
System.out.println("Info about manager");
m.managerdetails();
System.out.println("Info about officer");
o.officerdetails();
o.printsalary();
}
}
```

CLASS DIAGRAM



OUTPUT



RESULT

The program was successfully executed and familiarized method to print inheritance class in java.

EXPT. NO. 8	**TO CREATE AN ABSTRACT CLASS SHAPE**	DATE: 26-09-2020
Filename : Abstractshape.java	Compiler:javac	

AIM

To create an abstract class shape that contains an empty method named numberofsides().

ALGORITHM

Step 1:Start
Step 2:Create the abstract class shape that contains an empty method named numberof sides().
Step 3:Provide three classes named rectangle, triangle and hexagon such that each one of the classes extends the class shape.
Step 4:Each of the inherited class from shape class should provide the implementation for the method numberofsides().
Step 5:In the Abstractshape class, create the objects for the method numbrofsides().
Step 6:Create a super class reference variable and assign each child class object and invoke the methods and display the number of sides of different shapes.
Step 7:Stop.

PROGRAM

```
abstract class shape
{
public abstract void numberofsides();
}
class rectangle extends shape
{
public void numberofsides()
{
System.out.println("Number of sides=4");
}
}
class triangle extends shape
{
public void numberofsides()
{
System.out.println("Number of sides=3");
```

```
}
}
class hexagon extends shape
{
public void numberofsides()
{
System.out.println("Number of sides=6");
}
}
class Abstractshape
{
public static void main(String args[])
{
rectangle r=new rectangle();
triangle t=new triangle();
hexagon h=new hexagon();
shape sh;
sh=r;
sh.numberofsides();
sh=t;
sh.numberofsides();
sh=h;
sh.numberofsides();
}
}
```

OUTPUT

```
C:\Users\LENOVO I3\Desktop\my java>javac Abstractshape.java

C:\Users\LENOVO I3\Desktop\my java>java Abstractshape
Number of sides=4
Number of sides=3
Number of sides=6
```

RESULT

The program was successfully executed and familiarized method to print abstract in java.

EXPT. NO.9	**TO CREATE A CLASS FIGURE AND USE METHOD OVERLOADING TO CALCULATE THE AREA OF VARIOUS SHAPES.**	DATE: 18-10-2020
Filename : areashape.java	Compiler:javac/gdb	

AIM

To create a class Figure and use method overloading to calculate the area of various shapes.

ALGORITHM

ALGORITHM FOR CLASS Figue

Step 1: Create a class figure.
Step 2: Define method circlearea with radius and calculate the area of circle.
Step 3: Define method trianglearea with base and height and calculate the area of triangle.
Step 4: Define method rectanglearea with length and breadth and calculate the area of rectangle.

ALGORITHM FOR CLASS areashape

Step 1: Declare the main method.
Step 2: Inside main, create objects of class Figure.
Step 3: Invoke necessary methods of Figure class to pass the values to the methods.

PROGRAM

```
class Figure
{
void area(double r)
{
double area=3.14*r*r;
System.out.println("The area of circle="+area);
}
void area(double b,double h)
{
System.out.println("The area of triangle="+(b*h)/2);
}
void area(int a,int b)
{
```

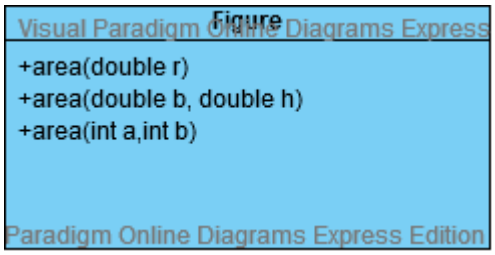
```
System.out.println("The area of rectangle="+a*b);
}
}
class areashapea
{
public static void main(String args[])
{
Figure f=new Figure();
f.area(30.5);
f.area(10.5,20.5);
f.area(10,5);
}
}
```

OUTPUT

```
C:\Users\LENOVO I3\Desktop\my java>javac areashape.java

C:\Users\LENOVO I3\Desktop\my java>java areashape
The area of circle=2920.985
The area of triangle=107.625
The area of rectangle=50
```

CLASS DIAGRAM



RESULT

The program was successfully executed and familiarized method overloading in java.

EXPT. NO. 10	**TO CREATE A SINGLE OBJECT TO CALL SUPERCLASS METHOD AND SUBCLASS METHOD**	DATE: 18-10-2020
Filename : emp.java	Compiler:javac	

AIM

To create a single object to call superclass method and subclass method.

ALGORITHM

ALGORITHM FOR CLASS Employee

Step 1:Create a class Employee.
Step 2:Define a method display() to display the class name.
Step 3:Define a method calcsalary() to display the salary of employee.
Step 4:Create a class engineer that inherits Employee.
Step 5:Define a method calcsalary() to display the salary of engineer.
Step 6:Call the methods of class Employee to class engineer.

ALGORITHM FOR CLASS emp

Step 1:Createthe main method.
Step 2:Create the object for class engineer.
Step 3:Invoke the methods in class engineer.

PROGRAM

```
class Employee
{
void display()
{
System.out.println("Name of the class is Employee");
}
void calcsalary()
{
System.out.println("Salary of employee is 10000");
}
}
class engineer extends Employee
{
```

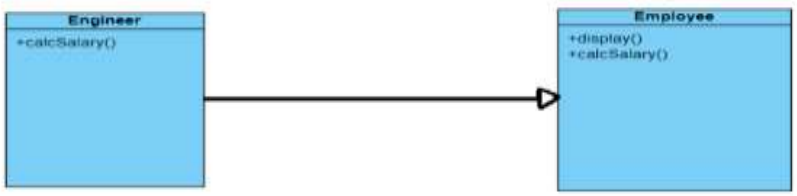
```
void calcsalary()
{
super.calcsalary();
System.out.println("Salary of engineer is 20000");
}
}
class emp
{
public static void main(String args[])
{
engineer e=new engineer();
e.display();
e.calcsalary();
}
}
```

OUTPUT

```
C:\Users\LENOVO I3\Desktop\my java>javac emp.java

C:\Users\LENOVO I3\Desktop\my java>java emp
Name of the class is Employee
Salary of employee is 10000
Salary of engineer is 20000
```

CLASS DIAGRAM



RESULT

The program was successfully executed and familiarized method to print superclass and subclass methods using single object in java.

EXPT. NO.11	**FILE HANDLING PROGRAM WITH READER/WRITER**	DATE: 22-10-2020
Filename : IOStreamExample .java	Compiler:javac/gdb	

AIM

To write a program to handle file using reader/writer.

ALGORITHM

1. Import the I/O classes.
2. Declare the class IOStreamsExample.
3. Declare the main method and exception.
4. Create FileReader object with file path.
5. Read data from file say hello.txt.
6. Write data to another file say mrng.txt.

PROGRAM

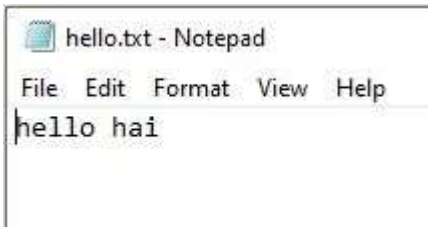
```
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
public class IOStreamsExample
{
public static void main(String args[])throws IOException
{
File file=new File("C:/Users/LENOVO I3/Desktop/my java/hello.txt");
FileReader reader=new FileReader(file);
char chars[]=new char[(int)file.length()];
reader.read(chars);
File out=new File("C:/Users/LENOVO I3/Desktop/my java/mrng.txt");
FileWriter writer=new FileWriter(out);
writer.write(chars);
for(char ch:chars)
{
System.out.print(ch);
}
```

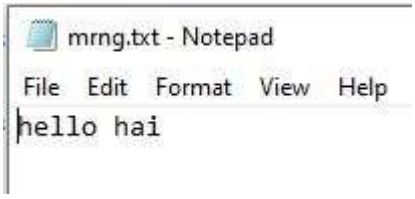
```
writer.flush();
System.out.println("Data successfully written in the specified file");
}
}
```

OUTPUT

```
C:\Users\LENOVO I3\Desktop\my java>javac IOStreamsExample.java

C:\Users\LENOVO I3\Desktop\my java>java IOStreamsExample
hello haiData successfully written in the specified file
```





RESULT

The program was successfully executed and familiarized method to print file using reader/writer in java.

EXPT. NO. 12	**TO READ FROM A FILE AND WRITE TO THE FILE BY HANDLING ALL FILE RELATED EXCEPTIONS**	DATE: 22-10-2020
Filename : copyfile.java	Compiler:javac	

AIM

To read from a file and write to the file by handling all file related exceptions.

ALGORITHM

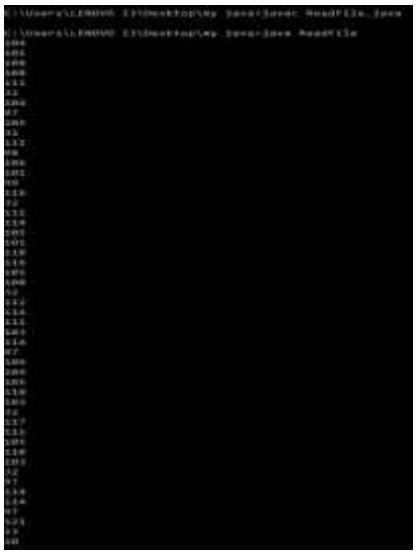
1. Declare class copyfile.
2. Declare the main method.
3. Declare Fileinputstream and FileoutputStream variables.
4. Create hello.txt and mrng.txt.
5. Create two files file1 and file2.
6. Handle the errors using try catch block.
7. Assign hello.txt to file1 and mrng.txt to file2.
8. Assign file1 and file2 to FIleInputStream and FileOutputStream variables.
9. Create a bye variable that stores the file1 values.
10. Read data from file say hello.txt.
11. Write data to another file say mrng.txt and print it in byte form.
12. In a final block declare the methods to close both the files.

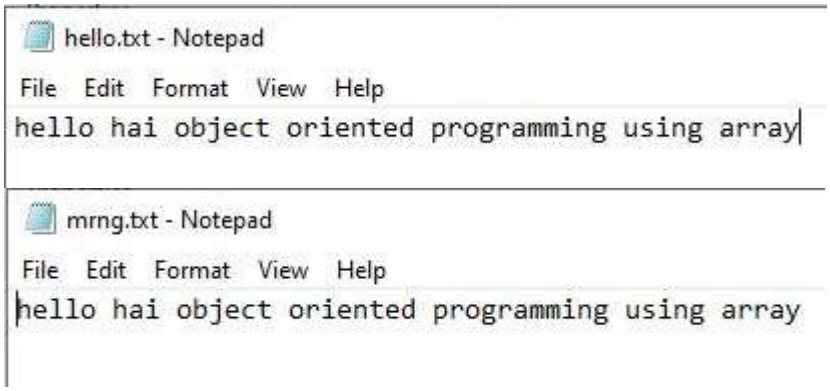
PROGRAM

```
import java.io.*;
class ReadFile
{
 public static void main(String args[]) throws IOException
{
  int i;
  FileInputStream fin = null;
  FileOutputStream fou = null;
  File file1,file2;
  byte bytes[];
try
{
 file1 = new File("hello.txt");
 file2 = new File("mrng.txt");
 fin = new FileInputStream(file1);
 fou = new FileOutputStream(file2);
 bytes = new byte[(int)file1.length()];
```

```
 fin.read(bytes);
 for(i=0;i<bytes.length;i++) {
    System.out.println(bytes[i]+"");
 }
 fou.write(bytes);
 fou.flush();
}
catch(IOException e)
{
 System.out.println("I/O Error: " + e);
}
finally
{
try
{
if(fin != null)
fin.close();
}
catch(IOException e2)
{
 System.out.println("Error Closing Input File");
}
try
{
if(fou != null)
 fou.close();
}
catch(IOException e2)
{
 System.out.println("Error Closing Output File");
}
}
}
}
```

OUTPUT





RESULT

The program was successfully executed and familiarized method to print files in java.

EXPT. NO. 13	**TO READ A LINE OF INTEGERS AND PRINT ITS SUM**	DATE: 22-10-2020
Filename : STDemo.java	Compiler:javac	

AIM

To read a line of integers and display it and print their sum.

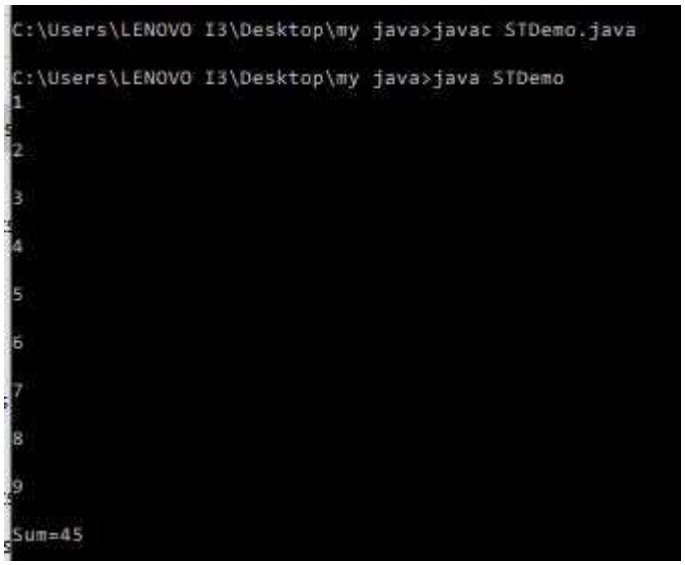
ALGORITHM

1. Declare the class STDemo.
2. Declare the input string.
3. Declare the main method.
4. Create the object of scanner class.
5. Create the object of StringTokenizer class.
6. Separate the string elements using StringTokenizer methods.
7. Find the sum of string elements.
8. Print the sum.

PROGRAM

```
import java.util.*;
class STDemo
{
static String in="1 2 3 4 5 6 7 8 9";
public static void main(String args[])
{
int n,sum=0;
Scanner sc=new Scanner(System.in);
StringTokenizer st=new StringTokenizer(in," ");
while(st.hasMoreTokens())
{
String key=st.nextToken();
System.out.println(key+"\n");
n=Integer.parseInt(key);
sum=sum+n;
}
System.out.println("Sum="+sum);
}
}
```

OUTPUT



RESULT

The program was successfully executed and familiarized with StringTokenizer in java.

EXPT. NO.14	**USAGE OF TRY, CATCH, THROWS AND FINALLY**	DATE: 31-10-2020
Filename : TestMyException.java	Compiler:javac/gdb	

AIM

To write a program to show the usage of try,catch, throws and finally.

ALGORITHM

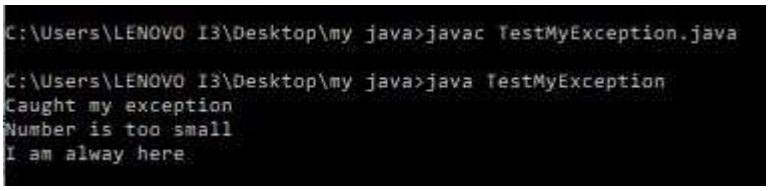
1. Start
2. Declare a class MyException that extends the class Exception.
3. Declare the constructor of the class with string parameter message.
4. Declare a class named TestMyException.
5. Declare the main method which throws an IllegalAccessException.
6. Declare and initialie two variables x and y.
7. Create a try block.
8. Evaluate the value of z using x and y from the following expression float z=(float)x/(float)y.
9. If z<0.01, use throw to print “the number is too small”.
10. Use catch keyword to catch the exception.
11. Declare a bloc using finally keyword to print a statement “I am here”.
12. Stop

PROGRAM

```
class MyException extends Exception
{
MyException(String message)
{
super(message);
}
}
class TestMyException
{
public static void main(String args[])throws IllegalAccessException
{
int x=5,y=1000;
try
{
```

```
float z=(float)x/(float)y;
if(z<0.01)
{
throw new MyException("Number is too small");
}
}
catch(MyException e)
{
System.out.println("Caught my exception");
System.out.println(e.getMessage());
}
finally
{
System.out.println("I am alway here");
}
}
}
```

OUTPUT



RESULT

The program was successfully executed and familiarized method to use try, catch, throws and finally

EXPT. NO. 15	**THREAD SYNCHRONIZATION**	DATE: 31-10-2020
Filename : copyfile.java	Compiler:javac	

AIM

To use thread synchronization.

ALGORITHM

1. Declare a class Callme.
2. Declare a method call with string parameter msg.
3. Print "[" with the content of msg.
4. Create a try block.
5. Call the sleep method from Thread class.
6. Use catch keyword to handle the exception.
7. Print "Interrupted" if there occurs any exception.
8. Close the try-catch block and print"]".
9. Create a class Caller that implements class Runnable.
10. Delclare the thread variable and call it with the constructor of class.
11. Declare a method run to print the statements in synchronized manner.
12. Use synchronized keyword to to access the parameters.
13. Declare class Synch1.
14. Declare the main method.
15. Create the object of Callme class.
16. Declare three objects for Caller class with the arguments to be passed.
17. In a try catch block call the method join() for every object of Caller class.
18. Use catch to handle the exception.

PROGRAM

```
class Callme
{
void call(String msg)
{
System.out.print("["+msg);
try
{
Thread.sleep(1000);
}
catch(InterruptedException e)
```

```
{
System.out.println("Interrupted");
}
System.out.println("]");
}
}
class Caller implements Runnable
{
String msg;
Callme target;
Thread t;
public Caller(Callme targ,String s)
{
target=targ;
msg=s;
t=new Thread(this);
t.start();
}
public void run()
{
synchronized(target)
{
target.call(msg);
}
}
}
class Synch1
{
public static void main(String args[])
{
Callme target=new Callme();
Caller ob1=new Caller(target,"Hello");
Caller ob2=new Caller(target,"Synchronized");
Caller ob3=new Caller(target,"World");
try
{
ob1.t.join();
ob2.t.join();
ob3.t.join();
}
catch(InterruptedException e)
{
System.out.println("Interrupted");
```

```
}
}
}
```

OUTPUT

RESULT

The program was successfully executed and familiarized with thread synchronization.

EXPT. NO.16	**JAVA SWING PROGRAM THAT WORKS AS CALCULATOR**	DATE: 7-11-2020
Filename : calculators.java	Compiler:javac/gdb	

AIM

To write a swing program that works as calculator.

ALGORITHM

13. Start
14. Declare a class calculators that extends the class JFrame and extends the interface ActionListener.
15. Declare the static variables.
16. Declare the constructor of the class.
17. Declare the main method.
18. Create a new frame/window using JFrame("calculator").
19. Create a try block.
20. Use setLookAndFeel function for the display of calculator.
21. Use catch keyword to catch the exception.
22. Create the object of the main class.
23. Use JTextField for setting the text component.
24. Call setEditable function and initialize it as false.
25. Create the buttons for operators and operands using JButton().
26. Create a panel to arrange the buttons using JPanel.
27. Register the buttons to perform action using addActionListener().
28. Add the buttons to the panel to perform the action using add().
29. Set the background colour to blue using setBackGround().
30. Add the button panel to the window using add().
31. Set the size of the window using setSize() and display it using show().
32. Declare a public method actionPerformed(ActionEvent e) to perform the operations.
33. Set the numbers to a string variable and delare it between 0 and 9.
34. Check each character of the string using charAt(), extract it and perform the required operation.
35. Stop

PROGRAM

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

```
class calculators extends JFrame implements ActionListener
{
static JFrame f;
static JTextField I;
String s0,s1,s2;
calculators()
{
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());
}
calculators c=new calculators();
I=new JTextField(16);
I.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("0");
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(I);
p.add(ba);
p.add(b1);
p.add(b2);
p.add(b3);
p.add(b4);
p.add(b5);
p.add(b6);
p.add(b7);
p.add(b8);
p.add(b9);
p.add(bm);
p.add(bs);
p.add(bd);
p.add(be);
p.add(b0);
p.add(beq);
p.add(beq1);
p.setBackground(Color.blue);
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)=='.')
```

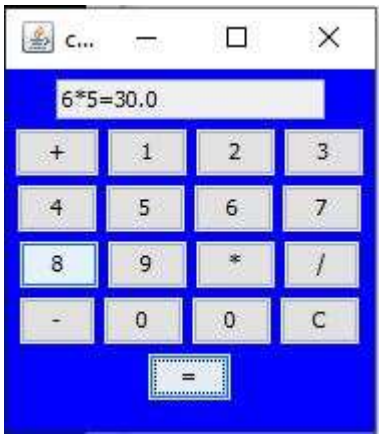
```
{
if(s1.equals(""))
s2=s2+s;
else
s0=s0+s;
I.setText(s0+s1+s2);
}
else if(s.charAt(0)=='C')
{
s0=s1=s2="";
I.setText(s0+s1+s2);
}
else if(s.charAt(0)=='=')
{
double te;
if(s1.equals("+"))
te=(Double.parseDouble(s0)+Double.parseDouble(s2));
else if(s1.equals("-"))
te=(Double.parseDouble(s0)-Double.parseDouble(s2));
else if(s1.equals("/"))
te=(Double.parseDouble(s0)/Double.parseDouble(s2));
else
te=(Double.parseDouble(s0)*Double.parseDouble(s2));
I.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="";
}
I.setText(s0+s1+s2);
}
}
}
```

OUTPUT

```
C:\Users\LENOVO I3\Desktop\my java krishna\lab>javac calculators.java
Note: calculators.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\LENOVO I3\Desktop\my java krishna\lab>java calculators
```



RESULT

The program was successfully executed and familiarized with the program.

EXPT. NO. 17	**PROGRAM TO STIMULATE A TRAFFIC SIGNAL**	DATE: 7-11-2020
Filename : traffic.java	Compiler:javac	

AIM

To write a program to stimulate a traffic signal.

ALGORITHM

19. Start.
20. Declare a class traffic that extends class Frame and implements ItemListener.
21. Declare the class variable clr.
22. Declare the constructor.
23. Create 3 Checkbox varibales,
24. Create the object for ChechboxGroup class.
25. Using the object name the checkbox variables.
26. Set the size of the frame using setSize() and title as traffic signal.
27. Set the layout to the center using setLayout and add the radio buttons to the center uing add().
28. Register the radio button using ItemListener().
29. Register the frame using WindowListener().
30. Declare a public method windowClosing(WindowEvent e).
31. Declare the main method.
32. Declare the public method itemStateChanged(ItemEvent e).
33. Initialize clr variable to provide the connection using e.getItem().
34. Declare the method paint() to draw and set colour.
35. Use drawString() to write traffic signals.
36. Use drawOval(),drawRect() to draw oval and rectangle for the traffic signals.
37. Use setColor() to set the colour and fillOval() to fill the desired colour to the oval.
38. Check the colour pressed and displayed with the if statement.
39. Stop

PROGRAM

```
import java.awt.*;
import java.awt.event.*;
class traffic extends Frame implements ItemListener
{
String clr="";
```

```
traffic()
{
Checkbox c1,c2,c3;
CheckboxGroup cbg=new CheckboxGroup();
c1=new Checkbox("red",true,cbg);
c2=new Checkbox("yellow",true,cbg);
c3=new Checkbox("green",true,cbg);
setSize(500,500);
setTitle("Traffic signal");
setVisible(true);
setLayout(new FlowLayout(FlowLayout.CENTER));
add(c1);
add(c2);
add(c3);
c1.addItemListener(this);
c2.addItemListener(this);
c3.addItemListener(this);
addWindowListener(new WindowAdapter()
{
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});
}
public static void main(String args[])
{
new traffic();
}
public void itemStateChanged(ItemEvent e)
{
clr=(e.getItem()).toString();
repaint();
}
public void paint(Graphics g)
{
g.drawString("Traffis signals",200,250);
g.drawOval(200,300,50,50);
g.drawOval(200,300,50,50);
g.drawOval(200,300,50,50);
g.drawRect(180,200,100,400);
if(clr.equals("red"))
{
```

```
g.setColor(Color.red);
g.fillOval(200,300,50,50);
}
if(clr.equals("green"))
{
g.setColor(Color.green);
g.fillOval(200,300,50,50);
}
if(clr.equals("yellow"))
{
g.setColor(Color.yellow);
g.fillOval(200,300,50,50);
}
}
}
```

OUTPUT

```
C:\Users\LENOVO I3\Desktop\my java krishna\lab>javac traffic.java

C:\Users\LENOVO I3\Desktop\my java krishna\lab>java traffic
```

RESULT

The program was successfully executed and familiarized with the program.

EXPT. NO. 18	**PROGRAM TO IMPLEMENT DOUBLY LINKED LIST**	DATE: 21-11-2020
Filename : DoublyLinkedList.java	Compiler:javac	

AIM

To program to implement doubly linked list.

ALGORITHM

1. Start.
2. Declare a class Node.
3. Declare the protected class variables.
4. Create the constructor.
5. Initialize the values to null.
6. Declare a constructor Node with parameters.
7. Declare the method setLinkNext().
8. Declare the method setLinkPrev().
9. Declare the node getLinkNext and getLinkPrev().
10. Declare the method getData().
11. Declare the method setData().
12. Declare class linkedlist.
13. Declare the protected variables.
14. Declare the constructor.
15. Declare a Boolean method isEmpty().
16. Declare the method getSize().
17. Declare the method insertAtStart(int val) to insert at front position.
18. Declare the method insertAtEnd(int val) to insert at end.
19. Declare the method insertAtPos(int val , int pos) to insert at any position.
20. Declare the method deleteAtPos(int pos) to delete at any position.
21. Declare the method display() the linked list.
22. Create a class DoublyLinkedList.
23. Declare the main method.
24. Create the Scanner class object.
25. Create the linkedlist object.
26. Using switch case call each operations according to user's wish.

PROGRAM

```
import java.util.Scanner;
class Node
{
protected int data;
protected Node next, prev;
public Node()
{
next = null;
prev = null;
data = 0;
}
public Node(int d, Node n, Node p)
{
data = d;
next = n;
prev = p;
}
public void setLinkNext(Node n)
{
next = n;
}
public void setLinkPrev(Node p)
{
prev = p;
}
public Node getLinkNext()
{
return next;
}
public Node getLinkPrev()
{
return prev;
}
public void setData(int d)
{
data = d;
}
public int getData()
{
return data;
```

```
}
}
class linkedList
{
protected Node start;
protected Node end ;
public int size;
public linkedList()
{
start = null;
end = null;
size = 0;
}
public boolean isEmpty()
{
return start == null;
}
public int getSize()
{
return size;
}
public void insertAtStart(int val)
{
Node nptr = new Node(val, null, null);
if(start == null)
{
start = nptr;
end = start;
}
else
{
start.setLinkPrev(nptr);
nptr.setLinkNext(start);
start = nptr;
}
size++;
}
public void insertAtEnd(int val)
{
Node nptr = new Node(val, null, null);
if(start == null)
{
start = nptr;
```

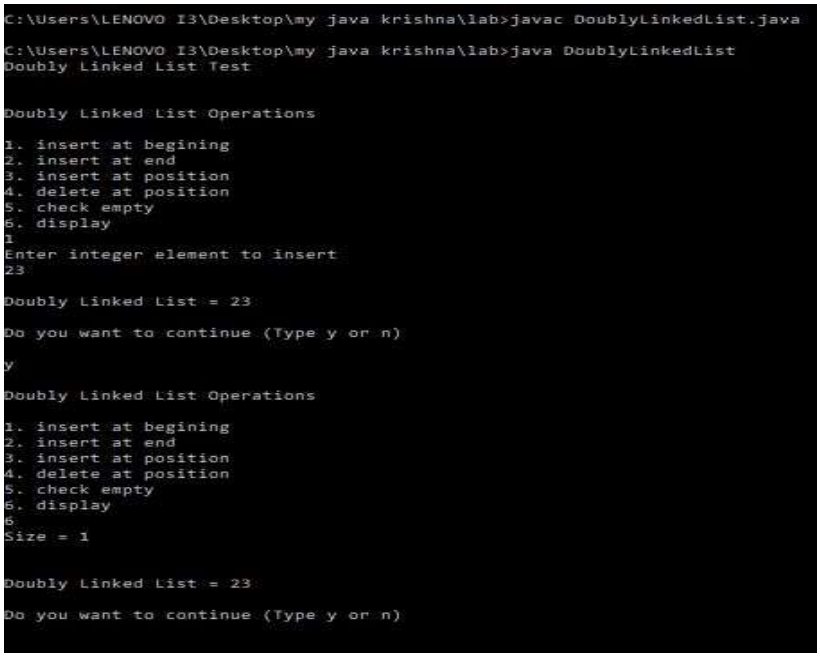
```
end = start;
}
else
{
nptr.setLinkPrev(end);
end.setLinkNext(nptr);
end = nptr;
}
size++;
}
public void insertAtPos(int val , int pos)
{
Node nptr = new Node(val, null, null);
if (pos == 1)
{
insertAtStart(val);
return;
}
Node ptr = start;
for (int i = 2; i <= size; i++)
{
if (i == pos)
{
Node tmp = ptr.getLinkNext();
ptr.setLinkNext(nptr);
nptr.setLinkPrev(ptr);
nptr.setLinkNext(tmp);
tmp.setLinkPrev(nptr);
}
ptr = ptr.getLinkNext();
}
size++ ;
}
public void deleteAtPos(int pos)
{
if (pos == 1)
{
if (size == 1)
{
start = null;
end = null;
size = 0;
return;
```

```
}
start = start.getLinkNext();
start.setLinkPrev(null);
size--;
return ;
}
if (pos == size)
{
end = end.getLinkPrev();
end.setLinkNext(null);
size-- ;
}
Node ptr = start.getLinkNext();
for (int i = 2; i <= size; i++)
{
if (i == pos)
{
Node p = ptr.getLinkPrev();
Node n = ptr.getLinkNext();
p.setLinkNext(n);
n.setLinkPrev(p);
size-- ;
return;
}
ptr = ptr.getLinkNext();
}
}
public void display()
{
System.out.print("\nDoubly Linked List = ");
if (size == 0)
{
System.out.print("empty\n");
return;
}
if (start.getLinkNext() == null)
{
System.out.println(start.getData() );
return;
}
Node ptr = start;
System.out.print(start.getData()+ " <-> ");
ptr = start.getLinkNext();
```

```
while (ptr.getLinkNext() != null)
{
System.out.print(ptr.getData()+ " <-> ");
ptr = ptr.getLinkNext();
}
System.out.print(ptr.getData()+ "\n");
}
}
public class DoublyLinkedList
{
public static void main(String[] args)
{
Scanner scan = new Scanner(System.in);
linkedList list = new linkedList();
System.out.println("Doubly Linked List Test\n");
char ch;
do
{
System.out.println("\nDoubly Linked List Operations\n");
System.out.println("1. insert at begining");
System.out.println("2. insert at end");
System.out.println("3. insert at position");
System.out.println("4. delete at position");
System.out.println("5. check empty");
System.out.println("6. display");
int choice = scan.nextInt();
switch (choice)
{
case 1 :
System.out.println("Enter integer element to insert");
list.insertAtStart( scan.nextInt() );
break;
case 2 :
System.out.println("Enter integer element to insert");
list.insertAtEnd( scan.nextInt() );
break;
case 3 :
System.out.println("Enter integer element to insert");
int num = scan.nextInt() ;
System.out.println("Enter position");
int pos = scan.nextInt() ;
if (pos < 1 || pos > list.getSize() )
System.out.println("Invalid position\n");
```

```
else
list.insertAtPos(num, pos);
break;
case 4 :
System.out.println("Enter position");
int p = scan.nextInt() ;
if (p < 1 || p > list.getSize() )
System.out.println("Invalid position\n");
else
list.deleteAtPos(p);
break;
case 5 :
System.out.println("Empty status = "+ list.isEmpty());
break;
case 6 :
System.out.println("Size = "+ list.getSize() +" \n");
break;
default :
System.out.println("Wrong Entry \n ");
break;
}
list.display();
System.out.println("\nDo you want to continue (Type y or n) \n");
ch = scan.next().charAt(0);
}
while (ch == 'Y'|| ch == 'y');
}
}
```

OUTPUT



```
C:\Users\LENOVO I3\Desktop\my java krishna\lab>javac DoublyLinkedList.java

C:\Users\LENOVO I3\Desktop\my java krishna\lab>java DoublyLinkedList
Doubly Linked List Test


Doubly Linked List Operations

1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. display
1
Enter integer element to insert
23

Doubly Linked List = 23

Do you want to continue (Type y or n)

y

Doubly Linked List Operations

1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. display
6
Size = 1


Doubly Linked List = 23

Do you want to continue (Type y or n)
```

RESULT

The program was successfully executed and familiarized with the program.

EXPT. NO.19	**JAVA PROGRAM TO IMPLEMENT QUICK SORT**	DATE: 21-11-2020
Filename : quick.java	Compiler:javac/gdb	

AIM

To write a swing program that works as calculator.

ALGORITHM

1. Create a class QuickSortOnStrings.
2. Declare the main method.
3. Create the object of QuickSortOnStrings class.
4. Initialise the array and array elemnts.
5. Call the sort method to sort the list.
6. Create a method called sort.
   6.1 if (array == null || array.length == 0)
   6.1.1 return
   6.2 this.names = array
   6.3 this.length = array.length
   6.4 Call the method quickSort
7. Declare the method quicksort.
   7.1 i = lowerIndex;
   7.2 j = higherIndex;
   7.3 String pivot = this.names[lowerIndex + (higherIndex - lowerIndex) / 2];
   7.4 Repaeat the steps for while (i <= j)
   7.5 Increment the value of i for (this.names[i].compareToIgnoreCase(pivot) < 0)
   7.6 Increment the value of j for (this.names[j].compareToIgnoreCase(pivot) > 0)
   7.7 Check if (i <= j) then
   7.7.1 exchangeNames(i, j)
   7.7.2 Increment the value of i.
   7.8 Decrement the value of j.
   7.9 Check if lowerIndex<j then
   7.9.1 call the method quickSort
   7.10 Check if i<higherIndex
   7.10.1 call the method quicksort

8 Declare a method exchangeNames(int i, int j)
   8.1 String temp = this.names[i]
   8.2 this.names[i] = this.names[j]

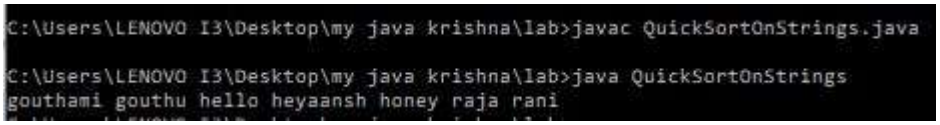
8.3 this.names[j] = temp

PROGRAM

```
public class QuickSortOnStrings
{
String names[];
int length;
public static void main(String[] args)
{
QuickSortOnStrings obj = new QuickSortOnStrings();
String stringsList[] = {"raja", "gouthu", "rani", "gouthami", "honey", "heyaansh", "hello"};
obj.sort(stringsList);
for (String i : stringsList)
{
System.out.print(i);
System.out.print(" ");
}
}
void sort(String array[])
{
if (array == null || array.length == 0)
{
return;
}
this.names = array;
this.length = array.length;
quickSort(0, length - 1);
}
void quickSort(int lowerIndex, int higherIndex)
{
int i = lowerIndex;
int j = higherIndex;
String pivot = this.names[lowerIndex + (higherIndex - lowerIndex) / 2];
while (i <= j)
{
while (this.names[i].compareToIgnoreCase(pivot) < 0)
{
i++;
}
while (this.names[j].compareToIgnoreCase(pivot) > 0) {
j--;
}
if (i <= j)
```

```
{
exchangeNames(i, j);
i++;
j--;
}
}
if (lowerIndex < j)
{
quickSort(lowerIndex, j);
}
if (i < higherIndex)
{
quickSort(i, higherIndex);
}
}
void exchangeNames(int i, int j) {
String temp = this.names[i];
this.names[i] = this.names[j];
this.names[j] = temp;
}
}
```

OUTPUT



RESULT

The program was successfully executed and familiarized with the program.

EXPT. NO. 20	**OPEN ENDED QUESTION**	DATE: 28-11-2020
Filename : rectinhsq.java	Compiler:javac/gdb	

AIM

To print area and perimeter of rectangle and square by inheriting the methods of class rectangle by class square.

ALGORITHM

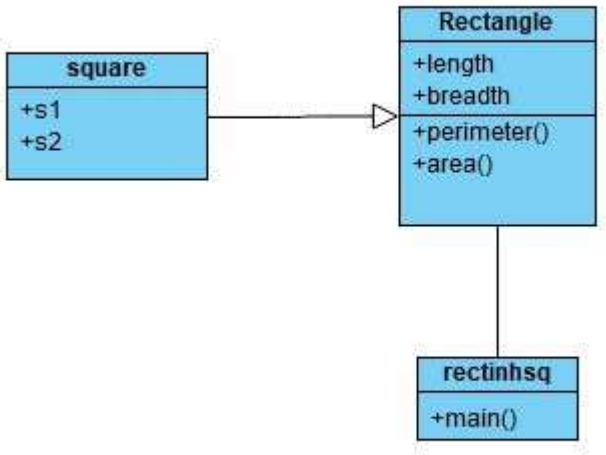
ALGORITHM FOR CLASS Rectangle

Step 1: Create a class Rectangle with instance variables length,breadth and s.
Step 2: Initialize the instance variable using parametrized constructors.
Step 3: Define method Rectangle to print the area and perimeter of the square.
Step 4: Define method Area and Perimeter to calculate the area and perimeter respectively.
Step 5: Create class Square that inherits the class Rectangle.
Step 6: Invoke method of parent class from child class.

ALGORITHM FOR CLASS rectinhsq

Step 1: Declare main method.
Step 2: Inside main,create objects of both Rectangle class and the child class Square with the required attributes , that are set in the constructor.
Step 3:Invoke necessary methods of Rectangle class to display the value of area and perimeter of rectangle and square

CLASS DIAGRAM



PROGRAM

```
import java.util.Scanner;
class rectangle
{
int length,breadth;
rectangle(int l,int b)
{
length=l;
breadth=b;
}
void perimeter()
{
int p=2*(length+breadth);
System.out.println("Perimeter="+p);
}
void area()
{
int a=length*breadth;
System.out.println("Area="+a);
}
```

```
}
class square extends rectangle
{
square(int s1,int s2)
{
super(s1,s2);
}
}
public class rectinhsq
{
public static void main(String args[])
{
Scanner sc=new Scanner(System.in);
int l,b,si;
System.out.println("Enter the length and breadth of rectangle");
l=sc.nextInt();
b=sc.nextInt();
System.out.println("Enter the side of square");
si=sc.nextInt();
square sq=new square(si,si);
System.out.println("Square\n");
sq.perimeter();
sq.area();
square r=new square(l,b);
System.out.println("Rectangle\n");
r.perimeter();
r.area();
}
}
```

OUTPUT

```
C:\Users\LENOVO I3\Desktop\my java>javac rectinhsq.java

C:\Users\LENOVO I3\Desktop\my java>java rectinhsq
Enter the length and breadth of rectangle
5
8
Enter the side of square
6
Square

Perimeter=24
Area=36
Rectangle

Perimeter=26
Area=40
```

RESULT

The program was successfully executed and familiarized method to print string in java.

EXPT. NO.21	**JAVA PROGRAM TO IMPLEMENT JAVA DATABASE CONNECTIVITY**	DATE: 7-12-2020
Filename : database.java	Compiler:javac/gdb	

AIM

To implement Database Connectivity using java.

ALGORITHM

Step 1: Start
Step 2: import java.sql package
Step 3: Declare a class name query
Step 4: Declare a main method
Step 5: Load and register the jdbc driver
Step 6: Open a connection to database
Step 7: Create a statement object to perform a query
Step 8: Execute the statement object and return the query resultset
Step 9: Process the resultset
Step 10: Repeat the steps while(result.next())
    10.1: print (result.getString(1))
    10.2:print (result.getString(2))
    10.3:print (result.getString(3))
Step 11: Stop

PROGRAM

```
import java.sql.*;   //import package
public class JdbcSelectTest
{
// Save as "JdbcSelectTest.java"
 public static void main(String[] args)  throws Exception
 {
  // register the driver
DriverManager.registerDriver(new sun.jdbc.odbc.JdbcOdbcDriver());
// Establish  Connection with database
Connection conn = DriverManager.getConnection( “jdbc:odbc:oradsn”,”scott”,”tiger”);
// Allocate a 'Statement' object in the Connection
```

```
Statement stmt = conn.createStatement();
// Execute a SQL  Statement  & result  is returned in a 'ResultSet' object.
 ResultSet rs = stmt.executeQuery("select * from emptab");
// Process the ResultSet by scrolling the cursor forward by next().
while(rs.next())
{
// Move the cursor to the next row, return false if no more row
System.out.println(rs.getInt(1));
System.out.println(rs.getString(2));
System.out.println(rs.getFloat(3));
}
// close the connection
conn.close();
}
}
```

RESULT

The program was successfully executed and familiarized with the program.