|  |  |  |
| --- | --- | --- |
| **DATE :** | **EX.NO:1** | **WRITE A JAVA PROGRAM TO CAPTURE THE PERSONAL INFORMATION OF A PERSON** |

**AIM:**

To write a java program to display the personal detail of a person

**ALGORITHM:**

|  |  |
| --- | --- |
| Step 1: | Import necessary header files |
| Step 2: | Create an appropriate class and main method |
| Step 3: | Receive the personal information as an input. |
| Step 4: | Display the personal details |
| Step 5: | Stop the program. |

**SOURCE CODE:**

import java.util.Scanner;

public class Personalinformation

{

public static void main(String args[]) throws Exception

{

String name, gender;

int age;

float weight;

Scanner SC=new Scanner(System.in);

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

name= SC.nextLine();

System.out.print("Enter Gender (Male/Female): ");

gender=SC.next();

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

age=SC.nextInt();

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

weight=SC.nextFloat();

System.out.println("Name: " + name);

System.out.println("Gender: " + gender);

System.out.println("Age: " + age);

System.out.println("Weight: " + weight);

}

}

**OUTPUT:**

**RESULT:**

|  |  |  |
| --- | --- | --- |
| **DATE :** | **EX.NO:2** | **DEVELOP AN APPLICATION USING INHERITANCE CONCEPT (SALARY PROGRAM)** |

**AIM:**

To write a java program to develop application for salary calculation using inheritance concept

**ALGORITHM:**

|  |  |
| --- | --- |
| Step 1: | Import necessary header files. |
| Step 2: | Create an Employee class with appropriate data members and member  functions to receive the data from the users and display the same . |
| Step 3: | Create other classes which extends Employee class ( eg: Programmer, Assistant Professor, Professor) with appropriate data members and member function required for calculating salary. |
| Step 4: | Create main function to calculate the salary by calling appropriate member function. |
| Step 5: | Stop the Program . |

**SOURCE CODE:**

Salary.java

import java.util.\*;

class Employee

{

int empid;

long mobile;

String name, address, mailid;

Scanner get = new Scanner(System.in);

void getdata()

{

System.out.println("Enter Name of the Employee");

name = get.nextLine();

System.out.println("Enter Mail id");

mailid = get.nextLine();

System.out.println("Enter Address of the Employee:");

address = get.nextLine();

System.out.println("Enter employee id ");

empid = get.nextInt();

System.out.println("Enter Mobile Number");

mobile = get.nextLong();

}

void display()

{

System.out.println("Employee Name: "+name);

System.out.println("Employee id : "+empid);

System.out.println("Mail id : "+mailid);

System.out.println("Address: "+address);

System.out.println("Mobile Number: "+mobile);

}

}

class Programmer extends Employee

{

double salary,bp,da,hra,pf,club,net,gross;

void getprogrammer()

{

System.out.println("Enter basic pay");

bp = get.nextDouble();

}

void calculateprog()

{

da=(0.97\*bp);

hra=(0.10\*bp);

pf=(0.12\*bp);

club=(0.1\*bp);

gross=(bp+da+hra);

net=(gross-pf-club);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"); System.out.println("PAY SLIP FOR PROGRAMMER");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"); System.out.println("Basic Pay: Rs. "+bp);

System.out.println("DA: Rs. "+da);

System.out.println("HRA: Rs. "+hra);

System.out.println("PF: Rs. "+pf);

System.out.println("CLUB: Rs. "+club);

System.out.println("GROSS PAY: Rs. "+gross);

System.out.println("NET PAY: Rs. "+net);

}

}

class Asstprofessor extends Employee

{

double salary,bp,da,hra,pf,club,net,gross;

void getasst()

{

System.out.println("Enter basic pay");

bp = get.nextDouble();

}

void calculateasst()

{

da=(0.97\*bp);

hra=(0.10\*bp);

pf=(0.12\*bp);

club=(0.1\*bp);

gross=(bp+da+hra);

net=(gross-pf-club);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("PAY SLIP FOR ASSISTANT PROFESSOR");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("Basic Pay: Rs. "+bp);

System.out.println("DA: Rs. "+da);

System.out.println("HRA: Rs. "+hra);

System.out.println("PF: Rs. "+pf);

System.out.println("CLUB: Rs. "+club);

System.out.println("GROSS PAY: Rs. "+gross);

System.out.println("NET PAY: Rs. "+net);

}

}

class Associateprofessor extends Employee

{

double salary,bp,da,hra,pf,club,net,gross;

void getassociate()

{

System.out.println("Enter basic pay");

bp = get.nextDouble();

}

void calculateassociate()

{

da=(0.97\*bp);

hra=(0.10\*bp);

pf=(0.12\*bp);

club=(0.1\*bp);

gross=(bp+da+hra);

net=(gross-pf-club);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("PAY SLIP FOR ASSOCIATE PROFESSOR");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("Basic Pay: Rs. "+bp);

System.out.println("DA: Rs. "+da);

System.out.println("HRA: Rs. "+hra);

System.out.println("PF: Rs. "+pf);

System.out.println("CLUB: Rs. "+club);

System.out.println("GROSS PAY: Rs. "+gross);

System.out.println("NET PAY: Rs. "+net);

}

}

class Professor extends Employee

{

double salary,bp,da,hra,pf,club,net,gross;

void getprofessor()

{

System.out.println("Enter basic pay");

bp = get.nextDouble();

}

void calculateprofessor()

{

da=(0.97\*bp);

hra=(0.10\*bp);

pf=(0.12\*bp);

club=(0.1\*bp);

gross=(bp+da+hra);

net=(gross-pf-club);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("PAY SLIP FOR PROFESSOR");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("Basic Pay: Rs. "+bp);

System.out.println("DA: Rs. "+da);

System.out.println("HRA: Rs. "+hra);

System.out.println("PF: Rs. "+pf);

System.out.println("CLUB: Rs. "+club);

System.out.println("GROSS PAY: Rs. "+gross);

System.out.println("NET PAY: Rs. "+net);

}

}

class Salary

{

public static void main(String args[])

{

int choice,cont;

do

{

System.out.println("PAYROLL");

System.out.println(" 1.PROGRAMMER \t 2.ASSISTANT PROFESSOR \t 3.ASSOCIATE PROFESSOR \t 4.PROFESSOR ");

Scanner c = new Scanner(System.in);

System.out.println("Enter Your Choice:");

choice=c.nextInt();

switch(choice)

{

case 1:

{

Programmer p=new Programmer();

p.getdata();

p.getprogrammer();

p.display();

p.calculateprog();

break;

}

case 2:

{

Asstprofessor asst=new Asstprofessor();

asst.getdata();

asst.getasst();

asst.display();

asst.calculateasst();

break;

}

case 3:

{

Associateprofessor asso=new Associateprofessor();

asso.getdata();

asso.getassociate();

asso.display();

asso.calculateassociate();

break;

}

case 4:

{

Professor prof=new Professor();

prof.getdata();

prof.getprofessor();

prof.display();

prof.calculateprofessor();

break;

}

}

System.out.print("Please enter 0 to quit and 1 to continue: ");

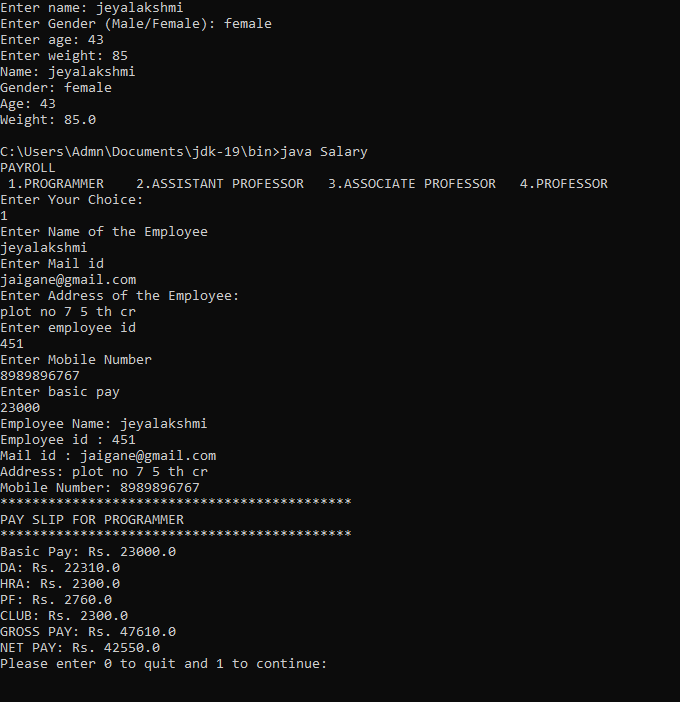
cont=c.nextInt();

}while(cont==1);

}

}

**OUTPUT:**



**RESULT:**

|  |  |  |
| --- | --- | --- |
| **DATE :** | **EX.NO:3** | **STRING OPERATIONS USING ARRAYLIST** |

**AIM:**

To write a java program for string operations using ArrayList

**ALGORITHM:**

|  |  |
| --- | --- |
| Step 1: | Import necessary header file |
| Step 2: | Create a class and main program to declare an Arraylist of strings |
| Step 3: | Perform differend string operation such as   1. Append a string to at the end of the existing list 2. Append a string at the specified index 3. Search a particular string in the array. 4. List string that starts with specified starting character 5. Size of the list and to remove the element in the list |
| Step 4: | Display the Output |
| Step 5: | Stop the Program |

**SOURCE CODE:**

Arraylistexample.java

import java.util.\*;

import java.io.\*;

public class Arraylistexample

{

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

{

ArrayList<String> obj = new ArrayList<String>();

DataInputStream in=new DataInputStream(System.in);

int c,ch;

int i,j;

String str,str1;

do

{

System.out.println("STRING MANIPULATION");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("1. Append at end \t 2.Insert at particular index \t 3.Search \t"); System.out.println("4. List string that starting with letter \t");

System.out.println("5. Size \t 6.Remove \t 7.Sort \t 8.Display\t" );

System.out.println("Enter the choice ");

c=Integer.parseInt(in.readLine());

switch(c)

{

case 1:

{

System.out.println("Enter the string ");

str=in.readLine();

obj.add(str);

break;

}

case 2:

{

System.out.println("Enter the string ");

str=in.readLine();

System.out.println("Specify the index/position to insert");

i=Integer.parseInt(in.readLine());

obj.add(i-1,str);

System.out.println("The array list has following elements:"+obj);

break;

}

case 3:

{

System.out.println("Enter the string to search ");

str=in.readLine();

j=obj.indexOf(str);

if(j==-1)

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

else

System.out.println("Index of:"+str+"is"+j);

break;

}

case 4:

{

System.out.println("Enter the character to List string that starts with specified character");

str=in.readLine();

for(i=0;i<(obj.size()-1);i++)

{

str1=obj.get(i);

if(str1.startsWith(str))

{

System.out.println(str1);

}

}

break;

}

case 5:

{

System.out.println("Size of the list "+obj.size());

break;

}

case 6:

{

System.out.println("Enter the element to remove");

str=in.readLine();

if(obj.remove(str))

{

System.out.println("Element Removed"+str);

}

else

{

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

}

break;

}

case 7:

{

Collections.sort(obj);

System.out.println("The array list has following elements:"+obj);

break;

}

case 8:

{

System.out.println("The array list has following elements:"+obj);

break;

}

}

System.out.println("Please Enter 0 to break and 1 to continue");

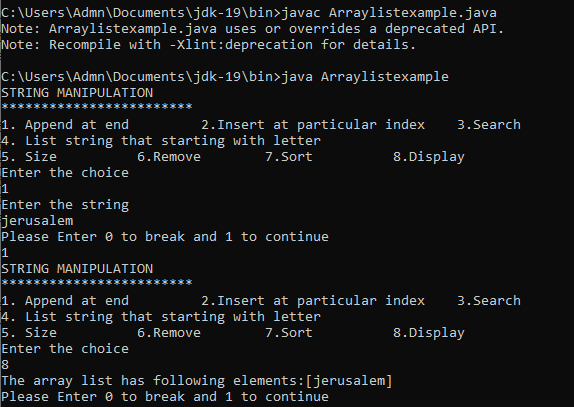
ch=Integer.parseInt(in.readLine());

}while(ch==1);

}

}

**OUTPUT:**



**RESULT :**

**OUTPUT:**

|  |  |  |
| --- | --- | --- |
| **DATE :** | **EX.NO:4** | **ARITHMETIC OPERATIONS WITH REQUIRED EXCEPTIONS HANDLING MECHANISM** |

**AIM:**

To write a java program to understand different arithmetic operators and exception handling mechanism

**ALGORITHM:**

|  |  |
| --- | --- |
| Step 1: | Import necessary header file |
| Step 2: | Create a class and main program to declare integer variables |
| Step 3: | Create a arithmetic exception – division by zero |
| Step 4: | Handle the exception using try- catch |
| Step 5: | Stop the Program |

**SOURCE CODE:**

class Example1

{

public static void main(String args[])

{

try{

int num1=30, num2=0;

int output=num1/num2;

System.out.println ("Result: "+output);

}

catch(ArithmeticException e){

System.out.println ("You Shouldn't divide a number by zero");

}

}

}

**RESULT:**

|  |  |  |
| --- | --- | --- |
| **DATE :** | **EX.NO:5** | **PROGRAM TO READ A FILE AND PRINT ON THE CONSOLE** |

**AIM:**

To write a java program to read a file and print the same as console output.

**ALGORITHM:**

|  |  |
| --- | --- |
| Step 1: | Import necessary header files. |
| Step 2: | Create a class and main program to declare integer variables |
| Step 3: | Create a arithmetic exception – division by zero |
| Step 4: | Handle the exception using try- catch |
| Step 5: | Stop the Program |

**SOURCE CODE:**

filedemo.java

import java.io.\*;

import java.util.\*;

class filedemo

{

public static void main(String args[])

{

String filename;

Scanner s=new Scanner(System.in);

System.out.println("Enter the file name ");

filename=s.nextLine();

File f1=new File(filename);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("FILE INFORMATION");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("NAME OF THE FILE "+f1.getName());

System.out.println("PATH OF THE FILE "+f1.getPath());

System.out.println("PARENT"+f1.getParent());

if(f1.exists())

System.out.println("THE FILE EXISTS ");

else

System.out.println("THE FILE DOES NOT EXISTS ");

if(f1.canRead())

System.out.println("THE FILE CAN BE READ ");

else

System.out.println("THE FILE CANNOT BE READ ");

if(f1.canWrite())

System.out.println("WRITE OPERATION IS PERMITTED");

else

System.out.println("WRITE OPERATION IS NOT PERMITTED");

if(f1.isDirectory())

System.out.println("IT IS A DIRECTORY ");

else

System.out.println("NOT A DIRECTORY");

if(f1.isFile())

System.out.println("IT IS A FILE ");

else

System.out.println("NOT A FILE");

System.out.println("File last modified "+ f1.lastModified());

System.out.println("LENGTH OF THE FILE "+f1.length());

System.out.println("FILE DELETED "+f1.delete());

}

}

**OUTPUT:**

**RESULT:**

|  |  |  |
| --- | --- | --- |
| **DATE :** | **EX.NO:6** | **MULTI THREADING** |

**AIM:**

To write a Java program to develop a multi-threaded application

**ALGORITHM:**

|  |  |
| --- | --- |
| Step 1: | Import necessary header file |
| Step 2: | Create a class that implements Runnable interface of Java |
| Step 3: | Create classes that extends thread class. |
| Step 4: | Create multiple threads from the classes created in Step3 |
| Step 5: | Create main class to instantiate the threads created at Step 4 |
| Step 6: | Kill the threads and end the execution |

**SOURCE CODE:**

multithreadprog.java

import java.util.\*;

class even implements Runnable

{

public int x;

public even(int x)

{

this.x = x;

}

public void run()

{

System.out.println("New Thread "+ x +" is EVEN and Square of " + x + " is: " + x \* x); }

}

class odd implements Runnable

{

public int x;

public odd(int x)

{

this.x = x;

}

public void run()

{

System.out.println("New Thread "+ x +" is ODD and Cube of " + x + " is: " + x \* x \* x); }

}

class A extends Thread

{

public void run()

{

int num = 0;

Random r = new Random();

try

{

for (int i = 0; i < 5; i++)

{

num = r.nextInt(100);

System.out.println("Main Thread and Generated Number is " + num);

if (num % 2 == 0)

{

Thread t1 = new Thread(new even(num));

t1.start();

}

else

{

Thread t2 = new Thread(new odd(num));

t2.start();

}

Thread.sleep(1000);

System.out.println("--------------------------------------");

}

}

catch (Exception ex)

{

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

}

}

}

public class multithreadprog

{

public static void main(String[] args)

{

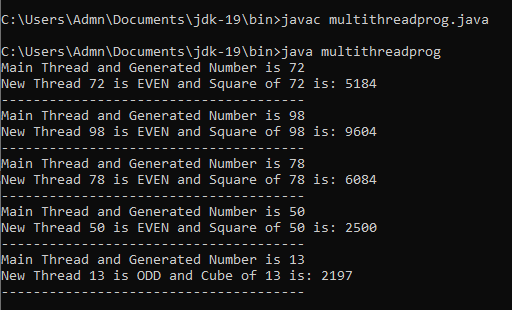
A a = new A();

a.start();

}

}

**OUTPUT:**



**RESULT:**

|  |  |  |
| --- | --- | --- |
| **DATE :** | **EX.NO:7** | **GENERIC FUNCTION** |

**AIM:**

To write a program to find the maximum and minimum value using generic function

**ALGORITHM:**

|  |  |
| --- | --- |
| Step 1: | Import necessary header file |
| Step 2: | Create a class with a template that extends comparable interface .Create appropriate constructor for the class .create two methods for comparison (eg: minimum, maximum) |
| Step 3: | Create main class to demonstrate the generic function created in Step2. |
| Step 4: | Call method function created in Step2 for comparison |
| Step 5: | Output the compared output for the objects created using generic function |
| Step 6: | Stop execution |

**SOURCE CODE:**

class MyClass<T extends Comparable<T>>

{

T[] vals;

MyClass(T[] o)

{

vals = o;

}

public T min()

{

T v = vals[0];

for(int i=1; i < vals.length; i++)

if(vals[i].compareTo(v) < 0)

v = vals[i];

return v;

}

public T max()

{

T v = vals[0];

for(int i=1; i < vals.length;i++)

if(vals[i].compareTo(v) > 0)

v = vals[i];

return v;

}

}

class genericdemo

{

public static void main(String args[]) {

int i;

Integer inums[]={10,2,5,4,6,1};

Character chs[]={'v','p','s','a','n','h'};

Double d[]={20.2,45.4,71.6,88.3,54.6,10.4};

MyClass<Integer> iob = new MyClass<Integer>(inums);

MyClass<Character> cob = new MyClass<Character>(chs);

MyClass<Double>dob = new MyClass<Double>(d);

System.out.println("Max value in inums: " + iob.max());

System.out.println("Min value in inums: " + iob.min());

System.out.println("Max value in chs: " + cob.max());

System.out.println("Min value in chs: " + cob.min());

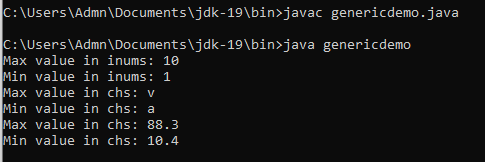
System.out.println("Max value in chs: " + dob.max());

System.out.println("Min value in chs: " + dob.min());

}

}

**OUTPUT:**



**RESULT:**

|  |  |  |
| --- | --- | --- |
| **DATE :** | **EX.NO:8** | **WORKING ON JAVA AWT** |

**AIM:**

To Design a login screen using JAVA AWT.

**ALGORITHM:**

|  |  |
| --- | --- |
| **Step 1:** | Import necessary header file for handling swing |
| **Step 2:** | Create a class that extends JFrame class and the interface ActionListener |
| **Step 3:** | Create appropriate panel buttons label, and text field as required |
| **Step 4:** | Add ActionListener for buttons and the action to be performed |
| **Step 5:** | Add all the components created in step 6 in the panel with appropriate layout |
| **Step 6:** | Validate the action of the button |
| **Step 7:** | Create new page for testing the same |
| **Step 8:** | Stop the Program |

**SOURCE CODE:**

**LoginFormDemo.java**

//import required classes and packages

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.lang.Exception;

//create CreateLoginForm class to create login form

//class extends JFrame to create a window where our component add

//class implements ActionListener to perform an action on button click

class CreateLoginForm extends JFrame implements ActionListener

{

//initialize button, panel, label, and text field

JButton b1;

JPanel newPanel;

JLabel userLabel, passLabel;

final JTextField textField1, textField2;

//calling constructor

CreateLoginForm()

{

//create label for username

userLabel = new JLabel();

userLabel.setText("Username"); //set label value for textField1

//create text field to get username from the user

textField1 = new JTextField(15); //set length of the text

//create label for password

passLabel = new JLabel();

passLabel.setText("Password"); //set label value for textField2

//create text field to get password from the user

textField2 = new JPasswordField(15); //set length for the password

//create submit button

b1 = new JButton("SUBMIT"); //set label to button

//create panel to put form elements

newPanel = new JPanel(new GridLayout(3, 1));

newPanel.add(userLabel); //set username label to panel

newPanel.add(textField1); //set text field to panel

newPanel.add(passLabel); //set password label to panel

newPanel.add(textField2); //set text field to panel

newPanel.add(b1); //set button to panel

//set border to panel

add(newPanel, BorderLayout.CENTER);

//perform action on button click

b1.addActionListener(this); //add action listener to button

setTitle("LOGIN FORM"); //set title to the login form

}

//define abstract method actionPerformed() which will be called on button click public void actionPerformed(ActionEvent ae) //pass action listener as a parameter {

String userValue = textField1.getText(); //get user entered username from the textField1

String passValue = textField2.getText(); //get user entered pasword from the textField2

//check whether the credentials are authentic or not

if (userValue.equals("test1@gmail.com") && passValue.equals("test")) { //if authentic, navigate user to a new page

//create instance of the NewPage

NewPage page = new NewPage();

//make page visible to the user

page.setVisible(true);

//create a welcome label and set it to the new page

JLabel wel\_label = new JLabel("Welcome: "+userValue);

page.getContentPane().add(wel\_label);

}

else{

//show error message

System.out.println("Please enter valid username and password"); }

}

}

//create the main class

class LoginFormDemo

{

//main() method start

public static void main(String arg[])

{

try

{

//create instance of the CreateLoginForm

CreateLoginForm form = new CreateLoginForm();

form.setSize(300,100); //set size of the frame

form.setVisible(true); //make form visible to the user

}

catch(Exception e)

{

//handle exception

JOptionPane.showMessageDialog(null, e.getMessage());

}

}

}

NewPage.java

//import required classes and packages

import javax.swing.\*;

import java.awt.\*;

//create NewPage class to create a new page on which user will navigate

class NewPage extends JFrame

{

//constructor

NewPage()

{

setDefaultCloseOperation(javax.swing.

WindowConstants.DISPOSE\_ON\_CLOSE);

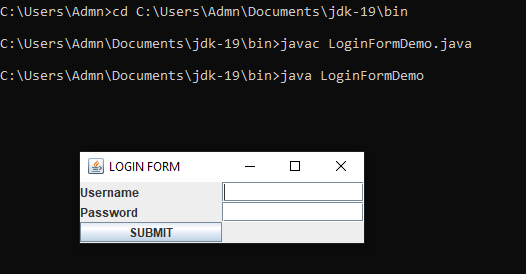
setTitle("Welcome");

setSize(400, 200);

}

}

**OUTPUT:**



**RESULT:**