**ADVANCED JAVA LAB MANUAL**

**20IS62A**

PROGRAM 1:

Design and create a swing based simple calculator program to perform basic arithmetic operations.

package a1;

import javax.swing.\*;

import java.awt.event.\*;

class Calculator implements ActionListener

{

JFrame f;

JTextField t;

JButton b1,b2,b3,b4,b5,b6,b7,b8,b9,b0,bdiv,

bmul,bsub,badd,bdec,beq,bdel,bclr;

static double a=0,b=0,result=0;

static int operator=0;

Calculator()

{

f=new JFrame("Calculator");

t=new JTextField();

b1=new JButton("1");

b2=new JButton("2");

b3=new JButton("3");

b4=new JButton("4");

b5=new JButton("5");

b6=new JButton("6");

b7=new JButton("7");

b8=new JButton("8");

b9=new JButton("9");

b0=new JButton("0");

bdiv=new JButton("/");

bmul=new JButton("\*");

bsub=new JButton("-");

badd=new JButton("+");

bdec=new JButton(".");

beq=new JButton("=");

bclr=new JButton("Clear");

t.setBounds(30,40,280,30);

b7.setBounds(40,100,50,40);

b8.setBounds(110,100,50,40);

b9.setBounds(180,100,50,40);

bdiv.setBounds(250,100,50,40);

b4.setBounds(40,170,50,40);

b5.setBounds(110,170,50,40);

b6.setBounds(180,170,50,40);

bmul.setBounds(250,170,50,40);

b1.setBounds(40,240,50,40);

b2.setBounds(110,240,50,40);

b3.setBounds(180,240,50,40);

bsub.setBounds(250,240,50,40);

bdec.setBounds(40,310,50,40);

b0.setBounds(110,310,50,40);

beq.setBounds(180,310,50,40);

badd.setBounds(250,310,50,40);

//bdel.setBounds(60,380,100,40);

bclr.setBounds(180,380,100,40);

f.add(t);

f.add(b7);

f.add(b8);

f.add(b9);

f.add(bdiv);

f.add(b4);

f.add(b5);

f.add(b6);

f.add(bmul);

f.add(b1);

f.add(b2);

f.add(b3);

f.add(bsub);

f.add(bdec);

f.add(b0);

f.add(beq);

f.add(badd);

f.add(bclr);

f.setLayout(null);

f.setVisible(true);

f.setSize(350,500);

//f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

f.setResizable(false);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

b5.addActionListener(this);

b6.addActionListener(this);

b7.addActionListener(this);

b8.addActionListener(this);

b9.addActionListener(this);

b0.addActionListener(this);

badd.addActionListener(this);

bdiv.addActionListener(this);

bmul.addActionListener(this);

bsub.addActionListener(this);

bdec.addActionListener(this);

beq.addActionListener(this);

//bdel.addActionListener(this);

bclr.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==b1)

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

if(e.getSource()==b2)

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

if(e.getSource()==b3)

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

if(e.getSource()==b4)

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

if(e.getSource()==b5)

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

if(e.getSource()==b6)

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

if(e.getSource()==b7)

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

if(e.getSource()==b8)

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

if(e.getSource()==b9)

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

if(e.getSource()==b0)

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

if(e.getSource()==bdec)

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

if(e.getSource()==badd)

{

a=Double.parseDouble(t.getText());

operator=1;

t.setText("");

}

if(e.getSource()==bsub)

{

a=Double.parseDouble(t.getText());

operator=2;

t.setText("");

}

if(e.getSource()==bmul)

{

a=Double.parseDouble(t.getText());

operator=3;

t.setText("");

}

if(e.getSource()==bdiv)

{

a=Double.parseDouble(t.getText());

operator=4;

t.setText("");

}

if(e.getSource()==beq)

{

b=Double.parseDouble(t.getText());

switch(operator)

{

case 1: result=a+b;

break;

case 2: result=a-b;

break;

case 3: result=a\*b;

break;

case 4: result=a/b;

break;

default: result=0;

}

t.setText(""+result);

}

if(e.getSource()==bclr)

t.setText("");

/\*if(e.getSource()==bdel)

{

String s=t.getText();

t.setText("");

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

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

}\*/

}

public static void main(String args[])

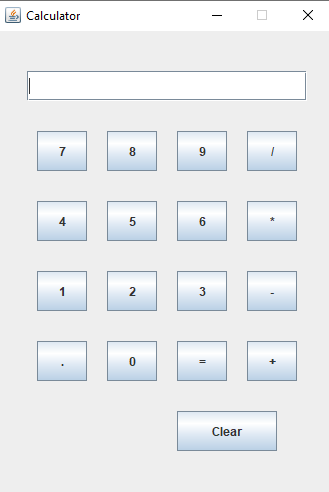
{

new Calculator();

}

}

OUTPUT:



Program 2:

Design and demonstrate loading of file in a Swing Component.

package a1;

import javax.swing.\*;

import java.awt.event.\*;

import java.io.\*;

public class filecoser extends JFrame implements ActionListener{

JMenuBar mb;

JMenu file;

JMenuItem open;

JTextArea ta;

filecoser()

{

open=new JMenuItem("Open File");

open.addActionListener(this);

file=new JMenu("File");

file.add(open);

mb=new JMenuBar();

mb.setBounds(0,0,800,20);

mb.add(file);

ta=new JTextArea(800,800);

ta.setBounds(0,20,300,300);

add(mb);

add(ta);

}

public void actionPerformed(ActionEvent e) {

if(e.getSource()==open){

JFileChooser fc=new JFileChooser();

int i=fc.showOpenDialog(this);

if(i==JFileChooser.APPROVE\_OPTION){

File f=fc.getSelectedFile();

String filepath=f.getPath();

System.out.println(filepath);

try{

FileReader fr=new FileReader(filepath);

BufferedReader br=new BufferedReader(fr);

//BufferedReader br=new BufferedReader(new FileReader(filepath));

String s1="",s2="";

while((s1=br.readLine())!=null)

{

s2+=s1+"\n";

}

ta.setText(s2);

br.close();

}

catch (Exception ex)

{

ex.printStackTrace();

}

} } }

public static void main(String[] args) {

filecoser f2=new filecoser();

f2.setSize(500,500);

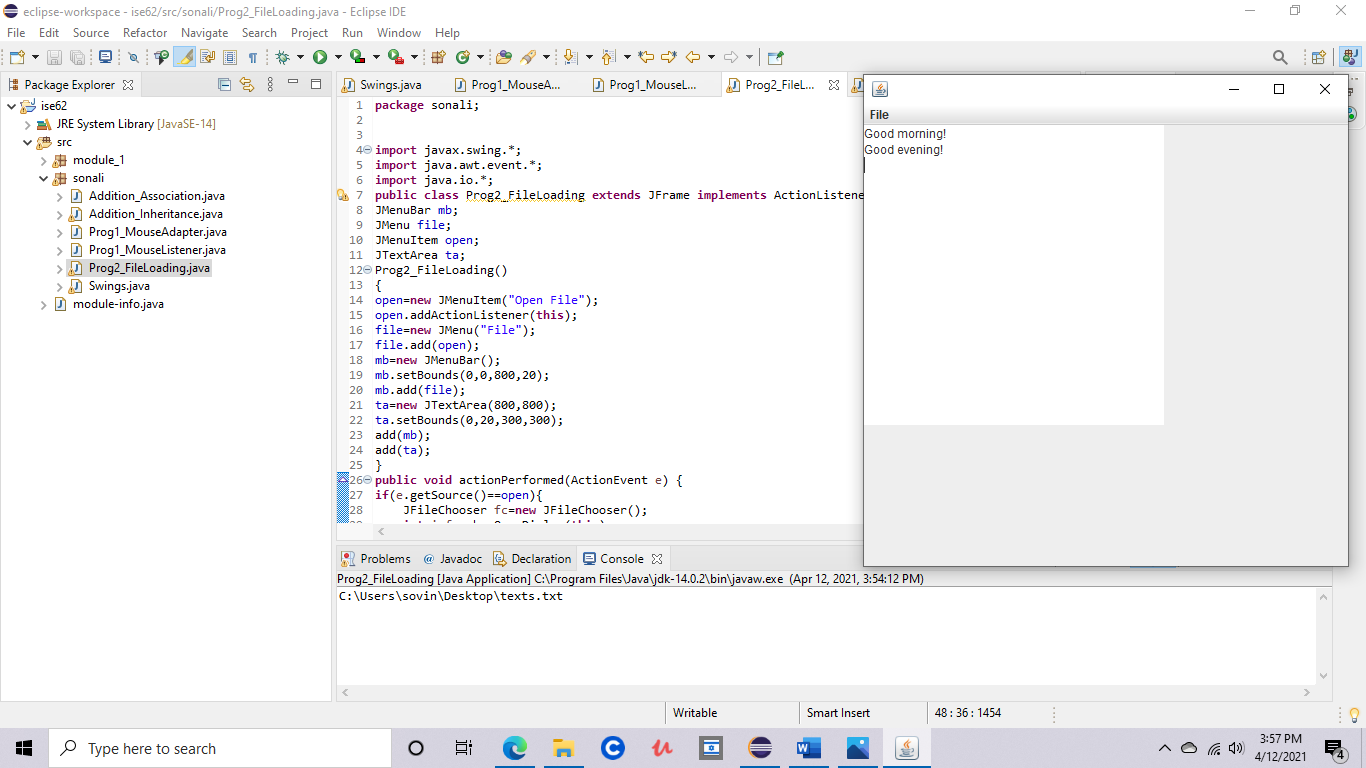
f2.setLayout(null);

f2.setVisible(true);

} }

OUTPUT:

OUTPUT:



Program 3:

Design and develop a swing based application to count the number of times a specific button is clicked by the user, apply event handling mechanism.

package prg3;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\*;

public class Prog3\_ClickCount implements ActionListener

{

JFrame f=new JFrame("welcome");

JButton b=new JButton("click");

JTextField t=new JTextField();

//JLabel l=new JLabel("hello");

int c=0;

Prog3\_ClickCount()

{

//l.setBounds(2,2,40,40);

t.setBounds(2,2,40,40);

b.setBounds(100,100,80,80);

f.setSize(300,300);

f.add(b);

f.add(t);

f.setLayout(null);

f.setVisible(true);

b.addActionListener(this);

//f.add(l);

}

public static void main(String args[])

{

new Prog3\_ClickCount();

}

@Override

public void actionPerformed(ActionEvent arg0)

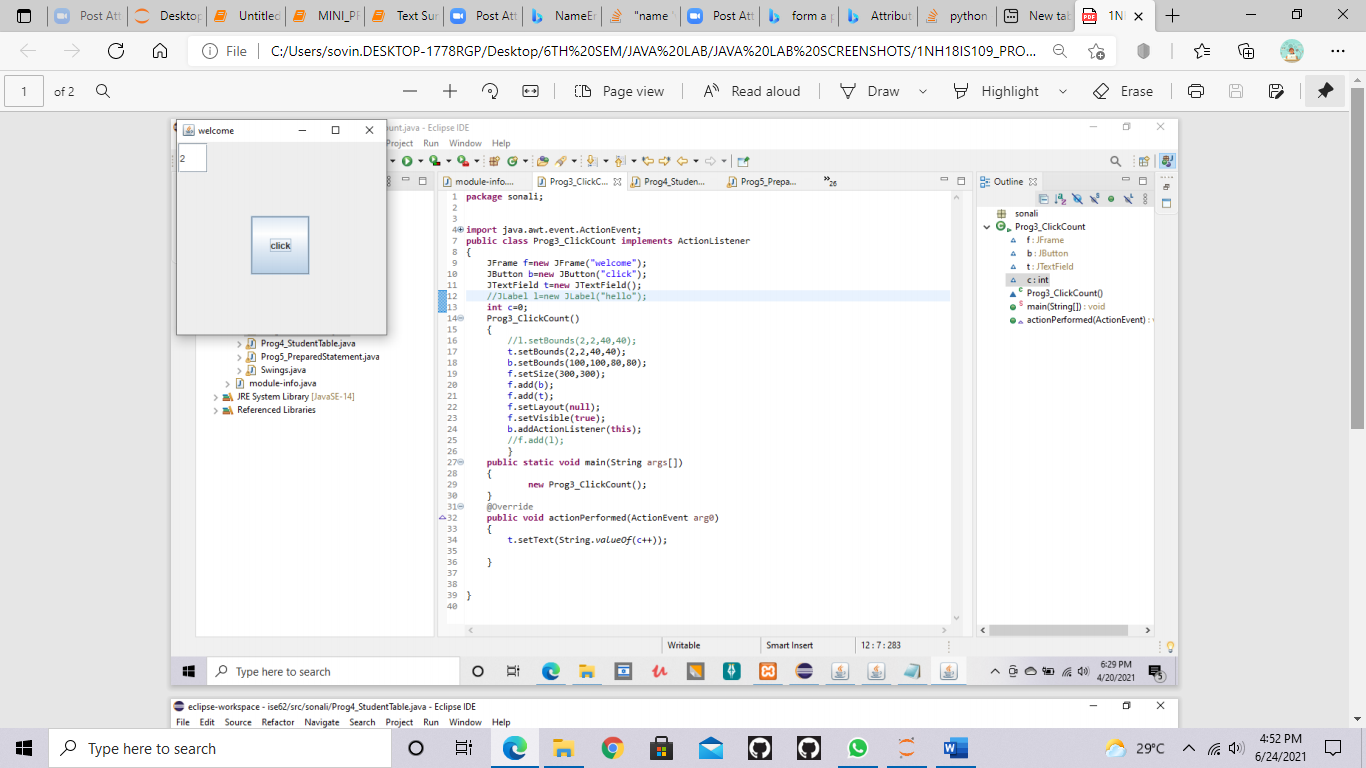
{

t.setText(String.valueOf(c++));

}

}

OUTPUT:



Program 4:

Design, Develop and Implement a JDBC program using statement object to display the Employee information to the console. Assume suitable columns and rows for the Employee table and JDBC drivers.

package a1;

import java.io.DataInputStream;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

public class empdb {

public static void main(String[] args) {

try{

Class.forName("com.mysql.cj.jdbc.Driver").newInstance();

Connection cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/","root","");

Statement smt=cn.createStatement();

smt.execute("CREATE DATABASE IF NOT EXISTS employee");

smt.execute("USE employee");

smt.execute("DROP TABLE IF EXISTS emp12");

smt.execute("CREATE TABLE emp12 ("+ "empid int(10) auto\_increment,"

+ "fname VARCHAR(25),"

+ "lname VARCHAR(25),"

+ "empdept VARCHAR(10),"

+"empdesn VARCHAR(10),"

+"empsal int(10),"

+ "PRIMARY KEY(empid)"

+ ")");

smt.execute("INSERT INTO emp12 (fname,lname,empdept,empdesn,empsal) VALUES" +

"('Ram','B','hr','mgr',5678),('jhon','ma','accnts','clk',6789),('Jill','Hill','quality','tester',56789)");

String q="Select \* from emp12";

ResultSet rs=smt.executeQuery(q);

while(rs.next())

{

System.out.println(rs.getString(1)+","+rs.getString(2)+","+rs.getString(3)+","+rs.getString(4)+","+rs.getString(5)+","+rs.getString(6));

}

cn.close();

}

catch(Exception e){

System.out.println(e);

}

}

}

OUTPUT:

1,Ram,B,hr,mgr,5678

2,jhon,ma,accnts,clk,6789

3,Jill,Hill,quality,tester,56789

Program 5:

Design, Develop and Implement a JDBC based program using Prepared statement object to perform update operation and display the updated Product data. Assume suitable columns for the Product table and JDBC drivers.

package a1;

import java.sql.\*;

import java.util.Scanner;

public class Pgm5{

public static void main(String[] args) {

try{

Class.forName("com.mysql.cj.jdbc.Driver").newInstance();

//serverhost = localhost, port=3306, username=root, password=123

Connection cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/","root","");

//insert statement

Statement s=cn.createStatement();

s.execute("CREATE DATABASE IF NOT EXISTS Product1");

s.execute("USE Product1");

s.execute("DROP TABLE IF EXISTS Prod12");

PreparedStatement p=cn.prepareStatement("CREATE TABLE Prod12("+"productid int(10) auto\_increment,"

+ "proname VARCHAR(25),"

+ "proddesc VARCHAR(25),"

+ "prodmodel VARCHAR(10),"

+"prodprice int(10),"

+ "PRIMARY KEY(productid)"

+ ")");

p.executeUpdate();

PreparedStatement prep = cn.prepareStatement("INSERT into Prod12(proname,proddesc,prodmodel,prodprice) VALUES (?,?,?,?) ");

prep.setString(1,"cds");

prep.setString(2,"store");

prep.setString(3,"c1");

prep.setInt(4,230);

int rowsInserted = prep.executeUpdate();

prep.setString(1,"cds1");

prep.setString(2,"store1");

prep.setString(3,"c11");

prep.setInt(4,2301);

prep.executeUpdate();

if (rowsInserted > 0) {

System.out.println("A new product is inserted successfully!");

}

//select

ResultSet res = prep.executeQuery("SELECT \* FROM Prod12");

while (res.next())

{

System.out.println(res.getString("proname") + "|" + res.getString("proddesc")+"|"+ res.getString("prodmodel")+"|"+res.getInt("prodprice"));

}

//update

PreparedStatement pst1 = cn.prepareStatement("UPDATE Prod12 SET prodprice=? WHERE proname=?");

pst1.setInt(1,2345);

pst1.setString(2,"cds");

int rowsUpdated = pst1.executeUpdate();

if (rowsUpdated > 0) {

System.out.println("An existing Product was updated successfully!");

}

//select to display changed values

ResultSet res1 = prep.executeQuery("SELECT \* FROM Prod12");

while (res1.next())

{

System.out.println(res1.getString("proname")+ "|" + res1.getString("proddesc")+"|"+ res1.getString("prodmodel")+"|"+res1.getInt("prodprice"));

}

pst1.setInt(1, 6789);

pst1.setString(2,"cds");

pst1.executeUpdate();

ResultSet res12 = prep.executeQuery("SELECT \* FROM Prod12");

while (res12.next())

{

System.out.println(res12.getString("proname")+ "|" + res12.getString("proddesc")+"|"+ res12.getString("prodmodel")+"|"+res12.getInt("prodprice"));

}}

catch (Exception e)

{

e.printStackTrace();

} }}

OUTPUT:

A new product is inserted successfully!

cds|store|c1|230

cds1|store1|c11|2301

An existing Product was updated successfully!

cds|store|c1|2345

cds1|store1|c11|2301

cds|store|c1|6789

cds1|store1|c11|2301

Program 6:

Design, Develop and Implement a JDBC based program using callable statement to execute a stored SQL procedure to display the details of a student matching the USN provided, from student table. Assume suitable columns and rows for the Student table and JDBC drivers.

**Procedure for Execution:**

Create a database with the name “test”

Create a table with the name “student” with columns “stdusn”(varchar), “stdname”,(varchar), “stdavg”,(integer).

Insert into the above table values first row: (1234,abcd,78), second row: (1235,abcd1,98).

Steps to create a procedure:

1. Click on the database (on test)
2. Click on Routines
3. Click on Add routine
4. Give the routine name as proc1
5. Type is PROCEDURE
6. In parameters :

First Parameter : Select IN from drop down, give name as value1, Length: 10

Second Paramter : Select OUT from dropdown, give name as value, Length: 10

1. In Definition type the below:

select count(stdusn) into value from student where stdusn=value1

1. Click on Go

Using the above steps a procedure is created with one input parameter and one more output parameter.

package a1;

import java.sql.\*;

import com.mysql.cj.jdbc.CallableStatement;

public class program6

{

public static void main(String[] args)

{

try{

Class.forName("com.mysql.cj.jdbc.Driver").newInstance();

//serverhost = localhost, port=3306, username=root, password=123

Connection cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/test","root","");

CallableStatement statement = (CallableStatement)cn.prepareCall("{call proc1(?,?)}");

//statement.setInt("value",98);

statement.setInt("value1",1235);

statement.registerOutParameter("value",Types.INTEGER);

statement.execute();

System.out.print("value is"+statement.getInt("value"));

System.out.println(" ");

ResultSet rs=statement.executeQuery("select \* from student");

while(rs.next())

{

System.out.println(rs.getString(1)+","+rs.getString(2)+","+rs.getString(3));

}

statement.close();

}

catch (Exception e)

{

e.printStackTrace();

}

}}

OUTPUT:

value is1

1234,abcd,78

1235,abcd1,90

1236,ram,78

1237,klop,98

1238,abdef,67

**PART-B**

Program 7:

Develop a library home page with option to search all books written by a particular author. Use servlet to make DB connectivity and JSP to display the list of books in tabular fashion to the client. Incase of error, JSP page must display error message to client. Assume suitable Data Base and drivers. Table can have columns author name and book title.

index.html

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<form action="http://localhost:9092/Libraryprogram/serv1" method="get"/>

<input type="text" name="authname"/>

<input type="submit" name="submit"/>

</body>

</html>

serv1.java(servlet)

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.Statement;

import java.sql.ResultSet;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/serv1")

public class serv1 extends HttpServlet {

private static final long serialVersionUID = 1L;

public serv1() {

super();

// TODO Auto-generated constructor stub

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

{

Connection connection = null;

Statement statement = null;

ResultSet resultSet = null;

String a1=request.getParameter("authname");

PrintWriter pw=response.getWriter();

pw.print(a1);

try{

Class.forName("com.mysql.cj.jdbc.Driver").newInstance();

connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/library", "root","");

statement=connection.createStatement();

String id=request.getParameter("authname");

String sql ="select \* from book where author\_name= '"+id+"' ";

boolean s1=statement.execute(sql);

resultSet = statement.executeQuery(sql);

request.setAttribute("data",resultSet);

if(resultSet.next())

{

RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/displaysuccess.jsp");

dispatcher.forward(request, response);

}

else

{

RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/failed.jsp");

dispatcher.forward(request, response);

}}

catch(Exception e){

System.out.println(e);

}}}

displaysuccess.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<%@page import="java.sql.ResultSet"%>

<%@page import="java.sql.Statement"%>

<%@page import="java.sql.Connection"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body><table border="1"><tr>

<td>author name</td>

<td>book title</td>

</tr>

<% ResultSet resultSet = null;%>

<%resultSet=(ResultSet)request.getAttribute("data"); %>

<%do {

%>

<tr>

<td><%=resultSet.getString("author\_name") %></td>

<td><%=resultSet.getString("book\_title") %></td>

</tr>

<%

}while(resultSet.next());

} catch (Exception e) {

e.printStackTrace();

%>

</table>

</body>

</html>

failed.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

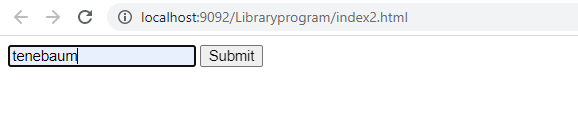
<body>

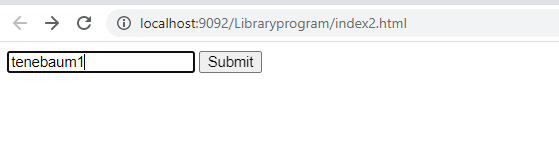
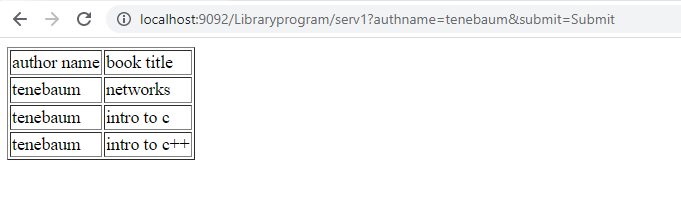
<p>NO SUCH AUTHOR</p>

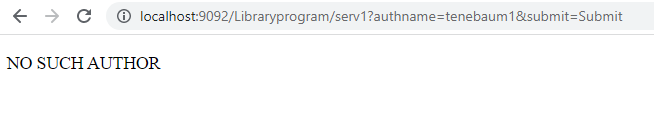
</body>

</html>

OUTPUT:







Program 8:

Design and develop a user login page and authenticate the user in a JSP page using database. Assume user name and password to be the column of the USER database. Establish connectivity using JDBC drivers.

index.html

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<form action="http://localhost:9092/userauthen/display.jsp" method="get"/>

<input type="text" name="username"/>

<input type="text" name="password"/>

<input type="submit" name="submit"/>

</body>

</html>

display.jsp

<%@page import="java.sql.DriverManager"%>

<%@page import="java.sql.ResultSet"%>

<%@page import="java.sql.Statement"%>

<%@page import="java.sql.Connection"%>

<%@ page errorPage="error.jsp" %>

<%String driver = "com.mysql.cj.jdbc.Driver";

String connectionUrl = "jdbc:mysql://localhost:3306/";

String database = "userlogin";

String userid = "root";

String password = "";

try {

Class.forName(driver);

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

Connection connection = null;

Statement statement = null;

ResultSet resultSet = null;

%>

<!DOCTYPE html>

<html>

<body>

<%=request.getParameter("username")%>

<%=request.getParameter("password")%>

<h1>Retrieve data from database in jsp</h1>

<table border="1">

<tr> <td>user name</td>

<td>password</td>

</tr>

<%

try {

connection = DriverManager.getConnection(connectionUrl+database, userid, password);

statement=connection.createStatement();

String id1=request.getParameter("username");

String id2=request.getParameter("password");

String sql ="select \* from login where username= '"+id1+"'&& password='"+id2+"'";

boolean sql1=statement.execute(sql);

out.print(sql1);

resultSet = statement.executeQuery(sql);%>

<% while(resultSet.next()){%>

<tr><td><%=resultSet.getString(1) %></td>

<td><%=resultSet.getString(2) %></td>

</tr>

<%

}

connection.close();

} catch (Exception e) {

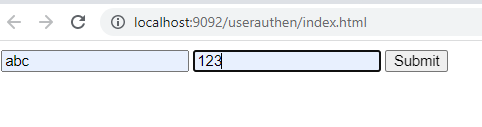
e.printStackTrace();

}

%>

</table> </body></html>

OUTPUT:





Program 9:

Create a HTML Page, which asks the user to enter a number in a textbox. On clicking the submit button, it places the request to a Servlet. The Servlet generates all Prime numbers which are less than the given number and adds them to an ArrayList and forwards the control to a JSP page. The JSP page iterates through the ArrayList and prints them in a tabular format. Apply RequestDispatcher methods to achieve the same.

NumberHome.html

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Servlet forwarding Program</title>

</head>

<body>

<form name="Form1" method="post"

action="http://localhost:9090/labpgms/WelcomeServlet">

<B>Enter a number:

<input type=textbox name="Number" size="25" value="">

<input type=submit value="Submit">

</form>

</body>

</html>

NumberDisplay.html

<%@page import="java.util.\*"%>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=windows-1255">

<title>Insert title here</title>

</head>

<body>

<h2>PrimeNumber list</h2>

<table>

<%

ArrayList<Integer> arr = (ArrayList<Integer>)request.getAttribute("primenumber");

%>

<% for(int i=1;i<arr.size();i++) { %>

<tr>

<td><% out.println(i);%> </td>

<td><% out.println(arr.get(i));%> </td>

</tr>

<% } %>

</table>

</body>

</html>

WelcomeServlet.java

import java.io.IOException;

import java.util.ArrayList;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/WelcomeServlet")

public class WelcomeServlet extends HttpServlet {

public WelcomeServlet() {

super();

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String str = request.getParameter("Number");

int number= Integer.parseInt(str);

ArrayList<Integer> returnarr= new ArrayList<Integer>();

returnarr= generatePrimeNumbers(number);

request.setAttribute("primenumber", returnarr);

RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/NumberDisplay.jsp");

dispatcher.forward(request, response);

}

ArrayList generatePrimeNumbers(int number)

{

ArrayList<Integer> arr = new ArrayList<Integer>();

for(int i=1; i <number; i++){

boolean isPrime = true;

for(int j=2; j < i ; j++){

if(i % j == 0){

isPrime = false;

break;

} }

// print the number

if(isPrime){

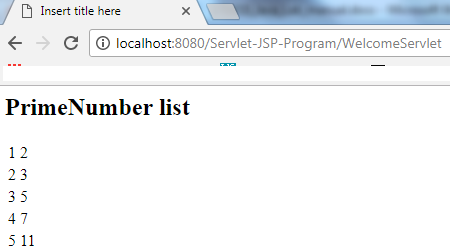
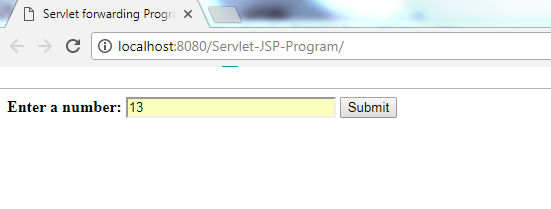
System.out.print(i + " ");

arr.add(i); }}

return arr;

}}

OUTPUT:



Program 10:

Design a Seminar registration form with Candidate Name, Email id, Favourite Courses enrolled. Apply usebean tag in JSP to receive and send the data to Candidate Java Bean and reply with successful registration and display the registered data.

index.html

<html>

<head>

<title>useBean action Test Page</title>

</head>

<body bgcolor="wheat">

<center><b>Register for Seminar</b></center>

<form action="http://localhost:9092/practicejsp/test.jsp" method="post">

<p>Please enter your User Name:

<input type="text" name="name"><br>

<p>please enter email id:

<input type="text" name="email"><br>

Favourite Courses Enrolled

<select name="language">

<option value="Java">Java

<option value="C++">C++

<option value="Perl">Perl

</select>

</p>

<p><input type="submit" value="Submit Information"></p>

</form>

</body>

</html>

LBean.java(class file)

package practicejsp;

public class LBean {

private String name;

private String email;

private String language;

private String languageComments;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getLanguage() {

return language;

}

public void setLanguage(String language) {

this.language = language;

}

public String getLanguageComments(){

if(language.equals("Java")){

return "The king of OO languages.";

}

else if(language.equals("C++")){

return "Rather too complex for some folk.";

}

else if(language.equals("Perl")){

return "OK,If you like comphrensible code";

}

else {

return "Sorry, I've never heard of " + language +".";

}

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

}

test.jsp

<jsp:useBean id="l1" class="practicejsp.LBean">

<jsp:setProperty name="l1" property="\*"/>

</jsp:useBean>

<html>

<head>

<title>useBean action Test Result</title>

</head>

<body bgcolor="wheat">

<p>Hello, name is

<jsp:getProperty name="l1" property="name" />

<p>Your Email id is<jsp:getProperty name="l1" property="email"/>

<p>Hello, language is

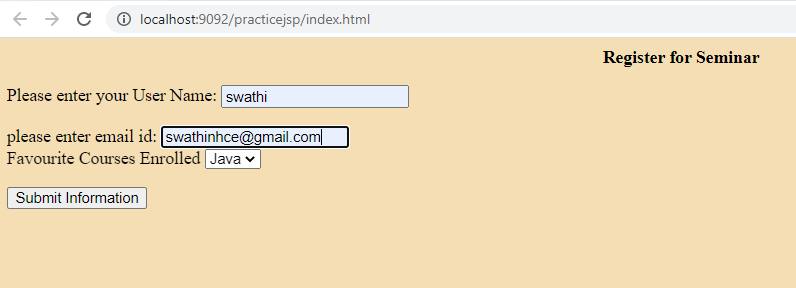
<jsp:getProperty name="l1" property="language" /></p>

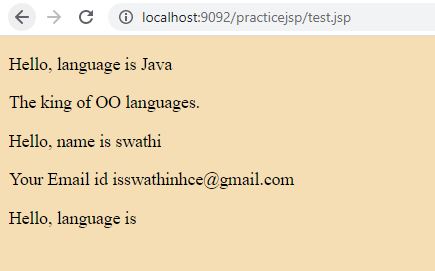
<p><jsp:getProperty name="l1" property="languageComments" /></p>

</body>

</html>

OUTPUT:





Program 11:

Create a Spring based examination conduction application to demonstrate the concept of dependency injection. Bean USER has the properties id, name, email id. Bean ANSWER has attributes id, answer, postedDate. Bean QUESTION has id, name, answer.

STEPS:

1. GOTO https://start.spring.io/
2. Give artifact name🡪click on generate (demo)
3. Zip file will be downloaded ,save the file in some location (ex: Desktop) and extract this zip
4. Goto Eclipse 🡪import🡪select Existing MAVEN🡪 Browse this extracted folder in to root location
5. Folder will be added in the explorer.
6. Goto com.example.demo 🡪 Run the demo1application.java
7. Further add all java files as class files as mentioned below.
8. Add .xml file in com.example.demo.resources location as an xml file.

DemoApplication.java

package com.example.demo;

import org.springframework.beans.factory.BeanFactory;

import org.springframework.beans.factory.xml.XmlBeanFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.core.io.ClassPathResource;

import org.springframework.core.io.Resource;

@SpringBootApplication

public class DemoApplication {

public static void main(String[] args) {

SpringApplication.run(DemoApplication.class, args);

Resource r=(Resource) new ClassPathResource("appc1.xml");

BeanFactory factory=new XmlBeanFactory((org.springframework.core.io.Resource) r);

User s=(User)factory.getBean("a1");

s.show();

}}

User.java

package com.example.demo;

public class User {

public User(String id, String emailid, String name,Question q1) {

super();

this.id = id;

this.emailid = emailid;

this.name = name;

this.q1=q1;

}

private String id;

private String emailid;

private String name;

private Question q1;

//private Answer a1;

public void show()

{

System.out.println("user id is"+id+"\n"+"user name is"+name+"\n"+"user email id is"+emailid+"\n");

System.out.println(q1);

}

}

Answer.java

package com.example.demo;

import java.sql.Date;

public class Answer {

public Answer(String aid, String answer, String posteddate) {

super();

this.aid = aid;

this.answer = answer;

this.posteddate = posteddate;

}

private String aid;

private String answer;

private String posteddate;

public String toString(){

return "answer id"+" "+aid+"\n"+"answer is"+answer+"\n"+"date is"+posteddate+"\n";

}

}

appc1.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans

xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:p="http://www.springframework.org/schema/p"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

<bean id="e" class="com.example.demo.Answer">

<constructor-arg value="a1" ></constructor-arg>

<constructor-arg value="userfunctions"></constructor-arg>

<constructor-arg value="23.4.2005"></constructor-arg>

</bean>

<bean id="e1" class="com.example.demo.Question">

<constructor-arg value="Q1"></constructor-arg>

<constructor-arg value="reusabiity"></constructor-arg>

<constructor-arg><ref bean="e"/></constructor-arg>

</bean>

<bean id="a1" class="com.example.demo.User">

<constructor-arg value="u1"></constructor-arg>

<constructor-arg value="user@gmail.com"></constructor-arg>

<constructor-arg value="user12"></constructor-arg>

<constructor-arg><ref bean="e1"/> </constructor-arg>

</bean>

</beans>

Question.java

package com.example.demo;

public class Question {

private String qid;

private String qname;

private Answer answer;

public Question(String qid, String qname, Answer answer) {

super();

this.qid = qid;

this.qname = qname;

this.answer = answer;

}

public String toString(){

return "question id is"+qid+"\n"+"question name is"+qname+"\n"+"answer is"+answer;

//return "anwer is"+answer;

}}

OUTPUT:

23:01:50.846 [main] DEBUG org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loaded 3 bean definitions from class path resource [appc1.xml]

23:01:50.863 [main] DEBUG org.springframework.beans.factory.xml.XmlBeanFactory - Creating shared instance of singleton bean 'a1'

23:01:50.905 [main] DEBUG org.springframework.beans.factory.xml.XmlBeanFactory - Creating shared instance of singleton bean 'e1'

23:01:50.908 [main] DEBUG org.springframework.beans.factory.xml.XmlBeanFactory - Creating shared instance of singleton bean 'e'

user id isu1

user name isuser12

user email id isuser@gmail.com

question id isQ1

question name isreusabiity

answer isanswer id a1

answer isuserfunctions

date is23.4.2005