**Practical 1a1 constructor overloading**

class Student {

int rollno;

String name;

static String college ="ITS";

Student(int r, String n) {

rollno = r;

name = n;

}

void display() {

System.out.println(rollno + " " + name + " " + college);

}

}

public class TestStaticVariable1 {

public static void main(String args[]) {

Student s1 = new Student(111, "Karan");

Student s2 = new Student(222, "Aryan");

s1.display();

s2.display();

}

}

**a2. Method Overloading**

public class sum

{

public int sum(int x, int y)

{

return(x+y);

}

public int sum(int x, int y,int z)

{

return(x+y+z);

}

public double sum(double x,double y)

{

return(x+y);

}

public static void main(String args[])

{

sum s=new sum();

System.out.println(s.sum(10,20));

System.out.println(s.sum(10,20,30));

System.out.println(s.sum(10.5,20.5));

}

}

**a3. Static method**

public class Student

{

int rollno;

String name,addr;

Student()

{

rollno=1001;

name="kiran Gowda";

addr="pawai";

}

Student(int r,String n,String a)

{

rollno=r;

name=n;

addr=a;

}

void display()

{

System.out.println("rollno \t name \t\t addr");

System.out.println(rollno+"\t"+name+"\t\t"+addr);

}

public static void main(String args[])

{

Student obj1=new Student();

Student obj2=new Student(1002,"mohan","thane");

obj1.display();

obj2.display();

}

}

**1b1. Single**

class Calculate {

int z;

public void addition(int x, int y)

{

z = x + y;

System.out.println("The sum of the given numbers: " + z);

}

public void subtraction(int x, int y)

{

z = x - y;

System.out.println("The difference between the given numbers: " + z);

}

}

class MyCalculate extends Calculate

{

public void multiplication(int x, int y)

{

z = x \* y;

System.out.println("The product of the given numbers: " + z);

}

}

public class Test {

public static void main(String args[]) {

int a = 20, b = 10;

MyCalculate demo = new MyCalculate();

demo.addition(a, b);

demo.subtraction(a, b);

demo.multiplication(a, b);

}

}

**b2. Multilevel**

class Car

{

public Car()

{

System.out.println("Class Car");

}

public void vehicleType()

{

System.out.println("Vehicle Type: Car");

}

}

class Maruti extends Car

{

public Maruti()

{

System.out.println("Class Maruti");

}

public void brand()

{

System.out.println("Brand: Maruti");

}

public void speed()

{

System.out.println("Max: 90Kmph");

}

}

public class Maruti800 extends Maruti

{

public Maruti800()

{

System.out.println("Maruti Model: 800");

}

public void speed()

{

System.out.println("Max: 80Kmph");

}

public static void main(String args[])

{

Maruti800 obj=new Maruti800();

obj.vehicleType();

obj.brand();

obj.speed();

}

}

**b3. Method Overriding**

class Bank

{

int getRateOfInterest()

{

return 0;

}

}

class SBI extends Bank

{

int getRateOfInterest()

{

return 8;

}

}

class ICICI extends Bank

{

int getRateOfInterest()

{

return 7;

}

}

class AXIS extends Bank

{

int getRateOfInterest()

{

return 9;

}

}

class Test2

{

public static void main(String args[])

{

SBI s=new SBI();

ICICI i=new ICICI();

AXIS a=new AXIS();

System.out.println("SBI Rate of Interest:"+s.getRateOfInterest());

System.out.println("ICICI Rate of Interest:"+i.getRateOfInterest());

System.out.println("AXIS Rate of Interest:"+a.getRateOfInterest());

}

}

**Practical 2a. Abstract classes and methods**

abstract class Animal

{

abstract void makeSound();

public void eat()

{

System.out.println("I can eat.");

}

}

class Dog extends Animal

{

public void makeSound()

{

System.out.println("Bark bark");

}

}

class Test3

{

public static void main(String[] args)

{

Dog d1=new Dog();

d1.makeSound();

d1.eat();

}

}

**2b. Concept on Interface**

interface Polygon

{

void getArea (int length, int breadth);

}

class Rectangle implements Polygon

{

public void getArea(int length,int breadth)

{

System.out.println("The area of the rectangle is"+(length\*breadth));

}

public static void main(String args[]) {

Rectangle r1=new Rectangle();

r1.getArea(5,6);

}

}

**b2. With multiple**

interface Backend {

public void connectServer();

}

class Fronted{

public void responsive(String str){

System.out.println(str+" can also be used to Fronted");

}

}

class Language extends Fronted implements Backend{

public void connectServer(){

System.out.println("jAVA CAN BE USED AS BACKEND");

}

public static void main(String args[]){

Language obj1=new Language();

obj1.responsive("java");

obj1.connectServer();

}

}

**Practical 3.1**

import java.util.Scanner;

public class Exception2 {

public static void main(String[] args)

{

int arr1[]=new int[10];

Scanner sc =new Scanner (System.in);

System.out.println("ENTER 10 ELEMENTS ONE BY ONE ");

for (int i=0;i<10;i++)

{

arr1[i]=sc.nextInt();

}

try

{

System.out.println("Array elements at 11 position"+arr1[11]);

}

catch(ArrayIndexOutOfBoundsException e)

{

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

}

}

}

**3.2**

import java.util.Scanner;

public class Exception3 {

public static void main(String[] args)

{

try

{

int a[]=new int[5];

a[10]=30/3;

}

catch(ArithmeticException e)

{

System.out.println("Arithmetic Exception occurs");

}

catch(ArrayIndexOutOfBoundsException e)

{

System.out.println("ArrayIndexOutOfBounds Exception occurs");

}

catch(Exception e)

{

System.out.println("Parent Exception occurs");

}

System.out.println("REST OF THE CODE.......!");

}

}

**3.3**

import java.util.Scanner;

public class HandlingExceptions

{

public static void main(String[] args)

{

int a, b, c = 0;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the first integer number");

a = sc.nextInt();

System.out.println("Enter the second integer number");

b = sc.nextInt();

try {

c = a / b;

} catch (ArithmeticException e) {

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

System.out.println("Denominator can't be zero");

}

System.out.println("Numerator is " + a);

System.out.println("Denominator is " + b);

System.out.println("Quotient is " + c);

}

}

**Practical 4a List interface**

import java.util.\*;

class ArrayListExamples {

public static void main(String args[]) {

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

al.add("Jack");

al.add("Smith");

Iterator<String> itr = al.iterator();

while (itr.hasNext()) {

System.out.println(itr.next());

}

}

}

**4b Set interface**

import java.util.\*;

class LinkedHashSetExample {

public static void main(String args[]) {

LinkedHashSet<String> al = new LinkedHashSet<String>();

al.add("Mariana");

al.add("Rick");

al.add("Sam");

Iterator<String> itr = al.iterator();

while (itr.hasNext()) {

System.out.println(itr.next());

}

}

}

**Practical 5**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

public class StudentResumeForms extends JFrame {

private JTextField nameField;

private JTextField addressField;

private JTextField emailField;

public StudentResumeForms() {

setTitle("Student Resume Form");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(400, 200);

JPanel panel = new JPanel(new GridLayout(3, 2));

JLabel nameLabel = new JLabel("Name:");

nameField = new JTextField(20);

JLabel addressLabel = new JLabel("Address:");

addressField = new JTextField(20);

JLabel emailLabel = new JLabel("Email:");

emailField = new JTextField(20);

JButton submitButton = new JButton("Submit");

submitButton.addActionListener((ActionEvent e) -> {

String name1 = nameField.getText();

String address = addressField.getText();

String email = emailField.getText();

// You can process and save this student information as needed.

System.out.println("Name: " + name1);

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

System.out.println("Email: " + email);

});

panel.add(nameLabel);

panel.add(nameField);

panel.add(addressLabel);

panel.add(addressField);

panel.add(emailLabel);

panel.add(emailField);

add(panel, BorderLayout.CENTER);

add(submitButton, BorderLayout.SOUTH);

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

StudentResumeForms form = new StudentResumeForms();

form.setVisible(true);

});

}

}

**Practical 7**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class SimpleCalculator extends JFrame {

private final JTextField textField;

private final JButton[] numberButtons = new JButton[10];

private final JButton addButton;

private final JButton subtractButton, multiplyButton, divideButton, equalsButton, clearButton;

private double firstNumber, secondNumber, result;

private char operator;

public SimpleCalculator() {

setTitle("Simple Calculator");

setSize(300, 400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

textField = new JTextField();

textField.setFont(new Font("Arial", Font.PLAIN, 20));

add(textField, BorderLayout.NORTH);

JPanel buttonPanel = new JPanel();

buttonPanel.setLayout(new GridLayout(4, 4));

for (int i = 0; i < 10; i++) {

numberButtons[i] = new JButton(String.valueOf(i));

numberButtons[i].setFont(new Font("Arial", Font.PLAIN, 20));

numberButtons[i].addActionListener(new NumberButtonListener());

buttonPanel.add(numberButtons[i]);

}

addButton = new JButton("+");

subtractButton = new JButton("-");

multiplyButton = new JButton("\*");

divideButton = new JButton("/");

equalsButton = new JButton("=");

clearButton = new JButton("C");

addButton.addActionListener(new OperatorButtonListener('+'));

subtractButton.addActionListener(new OperatorButtonListener('-'));

multiplyButton.addActionListener(new OperatorButtonListener('\*'));

divideButton.addActionListener(new OperatorButtonListener('/'));

equalsButton.addActionListener(new EqualsButtonListener());

clearButton.addActionListener(new ClearButtonListener());

buttonPanel.add(addButton);

buttonPanel.add(subtractButton);

buttonPanel.add(multiplyButton);

buttonPanel.add(divideButton);

buttonPanel.add(equalsButton);

buttonPanel.add(clearButton);

add(buttonPanel, BorderLayout.CENTER);

}

private class NumberButtonListener implements ActionListener {

@Override

public void actionPerformed(ActionEvent e) {

String buttonText = ((JButton) e.getSource()).getText();

textField.setText(textField.getText() + buttonText);

}

}

private class OperatorButtonListener implements ActionListener {

private final char op;

OperatorButtonListener(char operator) {

op = operator;

}

@Override

public void actionPerformed(ActionEvent e) {

firstNumber = Double.parseDouble(textField.getText());

operator = op;

textField.setText("");

}

}

private class EqualsButtonListener implements ActionListener {

@Override

public void actionPerformed(ActionEvent e) {

secondNumber = Double.parseDouble(textField.getText());

switch (operator) {

case '+' -> result = firstNumber + secondNumber;

case '-' -> result = firstNumber - secondNumber;

case '\*' -> result = firstNumber \* secondNumber;

case '/' -> {

if (secondNumber != 0) {

result = firstNumber / secondNumber;

} else {

textField.setText("Error");

return;

}

}

}

textField.setText(String.valueOf(result));

}

} private class ClearButtonListener implements ActionListener {

@Override

public void actionPerformed(ActionEvent e) {

textField.setText("");

firstNumber = 0;

secondNumber = 0;

result = 0;

}

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

SimpleCalculator calculator = new SimpleCalculator();

calculator.setVisible(true);

});

}

}

**Practical 8ab1**

<html>

<head>

<title>TODO supply a title</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<form name="frm1" method="post" action="Servlet1">

<label>Enter User Name</label><input type="text" name="txt1"/><br>

<input type="submit" name="click ME!"/>

</form>

</body>

</html>

**Practical 8ab2**

package pack1;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class Servlet1 extends HttpServlet

{

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException

{

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

String str1=request.getParameter("txt1");

Cookie ck=new Cookie("user",str1);

response.addCookie(ck);

out.print("<form action='Login'>");

out.print("<input type='submit' value='go'>");

out.print("</form>");

}

}

**Practical 8ab3**

package pack1;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class Login extends HttpServlet {

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException

{

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

Cookie ck[]=request.getCookies();

out.print("Hello "+ck[0].getValue()+"<br>");

out.print("Hello "+ck[0].getName());

}

}

**Practical 8c1**

<html>

<head>

<title>TODO supply a title</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<form name="frm1" method="post" action="Servlet1">

<label>Enter User Name</label><input type="text" name="txt1"/><br>

<input type="submit" name="click ME!"/>

</form>

</body>

</html>

**c2**

package pack1;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class Servlet1 extends HttpServlet

{

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException

{

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

String str1=request.getParameter("txt1");

HttpSession session=request.getSession();

session.setAttribute("name",str1);

out.print("<form action='Login'>");

out.print("<input type='submit' value='go'>");

out.print("</form>");

}

}

**c3**

package pack1;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class Login extends HttpServlet

{

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException

{

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

HttpSession session=request.getSession(false);

if(session!=null)

{

String name=(String)session.getAttribute("name");

out.print("Hello, "+name+" Welcome to the world of Servlet");

}

}

}

**Practical 9ab1**

<html>

<head>

<title>TODO supply a title</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

/head&gt;

<body>

<form action="RegServlet" method="post">

User name : <input type="text" name="uname"> <br>

Password :<input type="password" name="pwd"><br>

Email Id : <input type="text" name="email"> <br>

Country : <select name="coun">

<option>select...

<option> India

<option> Bangladesh

<option> Bhutan

<option>Canada

</select> &lt;br&gt;

<input type="submit" value="Register">

</form>

</body>

</html>

**ab2**

package pack1;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.sql.\*;

public class RegServlet extends HttpServlet

{

Connection con;

Statement st;

PreparedStatement pst;

ResultSet rs;

ResultSetMetaData rsmd;

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException

{

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

String str1,str2,str3,str4;

str1=request.getParameter("uname");

str2=request.getParameter("pwd");

str3=request.getParameter("email");

str4=request.getParameter("coun");

try

{

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

}

catch (ClassNotFoundException e)

{

}

try

{

String url="jdbc:mysql://localhost:3306/db1";

con=DriverManager.getConnection(url, "root","123456");

String query="insert into Reg values(?,?,?,?)";

pst=con.prepareStatement(query);

pst.setString(1,str1);

pst.setString(2,str2);

pst.setString(3,str3);

pst.setString(4,str4);

pst.executeUpdate();

st=con.createStatement();

rs=st.executeQuery("select \* from Reg");

rsmd=rs.getMetaData();

int cols=rsmd.getColumnCount();

out.println("<center><table border='1'>");

out.println("<tr>");

for(int i=1;i<=cols;i++)

{

out.println("<th>"+rsmd.getColumnName(i)+"</th>");

}

out.println("</tr>");

while(rs.next())

{ out.println("<tr><td>"+rs.getString(1)+"</td><td>"+rs.getString(2)+"</td><td>"+rs.getString(3)+"</td><t

d>"+rs.getString(4)+"</td></tr>");

}

out.println("</table></center>");

}

catch (Exception e)

{

}

}

}