# INDEX

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. no. | Practical | Date | Signature/Remark |
| 1 | 1. Write a program to create a class and implement the concepts of Constructor Overloading, Method Overloading and Static Methods 2. Write a program to create a class and implement the concepts of inheritance and Method Overriding |  |  |
| 2 | 1. Write a program to implement the concepts of Abstract classes and methods. 2. Write a program to implement the concepts of Interfaces |  |  |
| 3 | Write a program to define User-defined Exceptions and raise them as per the requirements. |  |  |
| 4 | Write a program to demonstrate the methods of:   1. List Interface 2. Set Interface 3. Map Interface |  |  |
| 5 | Write a program using various swing components to design Java applications to accept a student resume. |  |  |
| 6 | 1. Write a JDBC program that displays the data of a given table. 2. Write a JDBC program to return the data of a specific record from a given table. 3. Write a JDBC program to insert/update/delete records into a given table. |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | 1. Construct a simple calculator using the Java Swing with Minimum functionality. 2. Construct a GUI using Java Swings to accepts details of a record of a given table and submit it to the database using JDBC technology on the click of a button. |  |  |
| 8 | 1. Write a Servlet that accepts a username from a HTML form and stores it as a cookie. Write another Servlet that returns the value of this cookie and displays it. 2. Write a Servlet that displays the name and values of the cookies stored on the client. 3. Write a Servlet that accepts a Username from a HTML from and stores it as a session variable. Write another Servlet that returns the value of this session variable and display it. |  |  |
| 9 | 1. Write a registration servlet that accepts the data for a given table and stores it in the database. 2. Write a Servlet that displays all the records of a table. |  |  |
| 10 | 1. Write a JSP that accepts a username from a HTML form and stores it as a cookie. Write another JSP that returns the value of this cookie and displays it. 2. Write a JSP that displays the name and values of the cookies stored on the client. 3. Write a JSP that accepts a Username from a HTML from and stores it as a session variable. Write another JSP that returns the value of this session variable and displays it. |  |  |
| 11 | 1. Write a JSP code that accepts username and password from HTML file and validates the user from the database. 2. Write a registration JSP that accepts the data for a given table and stores it in the database. 3. Write a JSP that displays all the records of a table. |  |  |
| 12 | Write a Java application to encoding and decoding JSON in Java |  |  |

**Practical 1**

**Aim:** a) Write a program to create a class and implement the concepts of Constructor Overloading, Method Overloading, and Static Methods.

b) Write a program to create a class and implement the concepts of inheritance and Method Overriding

**Program (a):**

class MyClass {

private String name;

private int value;

// Constructor Overloading

public MyClass() {

this.name = "Default";

this.value = 0;

}

public MyClass(String name) {

this.name = name;

this.value = 0;

}

public MyClass(String name, int value) {

this.name = name;

this.value = value;

}

// Method Overloading

public void display() {

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

}

public void display(String prefix) {

System.out.println(prefix + " Name: " + name + ", Value: " + value);

}

// Static Method

public static void showInfo() {

System.out.println("This is a demonstration of Constructor Overloading, Method Overloading, and Static Methods.");

}

}

public class Main {

public static void main(String[] args) {

// Calling static method

MyClass.showInfo();

// Using Constructor Overloading

MyClass obj1 = new MyClass();

MyClass obj2 = new MyClass("John");

MyClass obj3 = new MyClass("Alice", 42);

// Using Method Overloading

obj1.display();

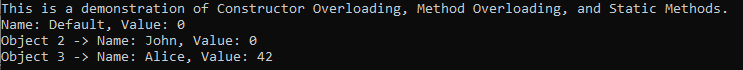
obj2.display("Object 2 ->");

obj3.display("Object 3 ->");

}

}

**Output:**

****

**Program (b):**

class Animal {

public void sound() {

System.out.println("Animals make sounds.");

}

public void eat() {

System.out.println("Animals eat food.");

}

}

class Dog extends Animal {

@Override

public void sound() {

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

}

@Override

public void eat() {

System.out.println("Dogs eat bones.");

}

}

public class Example {

public static void main(String[] args) {

Animal genericAnimal = new Animal();

genericAnimal.sound();

genericAnimal.eat();

Dog dog = new Dog();

dog.sound();

dog.eat();

Animal polymorphicDog = new Dog();

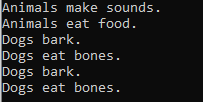
polymorphicDog.sound();

polymorphicDog.eat();

}

}

**Output:**

****

**Practical 2**

**Aim:** a) Write a program to implement the concepts of Abstract classes and methods. b) Write a program to implement the concepts of interfaces.

**Program (a):**

abstract class Shape {

abstract void draw(); // Abstract method

}

class Circle extends Shape {

@Override

void draw() {

System.out.println("Drawing a Circle.");

}

}

class Rectangle extends Shape {

@Override

void draw() {

System.out.println("Drawing a Rectangle.");

}

}

public class LOL {

public static void main(String[] args) {

Shape circle = new Circle();

Shape rectangle = new Rectangle();

circle.draw();

rectangle.draw();

}

}

**Output:**

****

**Program (b):**

interface Animal {

void sound();

}

class Dog implements Animal {

public void sound() {

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

}

}

class Cat implements Animal {

public void sound() {

System.out.println("Cats meow.");

}

}

public class Interface {

public static void main(String[] args) {

Animal dog = new Dog();

Animal cat = new Cat();

dog.sound();

cat.sound();

}

}

**Output:**

****

**Practical 3**

**Aim:** Write a program to define User-defined Exceptions and raise them as per the requirements.

**Program:**

class NegativeAgeException extends Exception {

public NegativeAgeException(String message) {

super(message);

}

}

class Person {

private int age;

// Set age with validation

public void setAge(int age) throws NegativeAgeException {

if (age < 0) {

throw new NegativeAgeException("Age cannot be negative.");

}

this.age = age;

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

}

}

public class L {

public static void main(String[] args) {

Person person = new Person();

try {

person.setAge(9);

person.setAge(-6);

} catch (NegativeAgeException e) {

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

}

}

}

**Output:**

****

**Practical 4**

**Aim:** Write a program to demonstrate the methods of:

a. List Interface

b. Set Interface

c. Map Interface

**Program:**

import java.util.\*;

public class InterfaceDemo {

public static void main(String[] args) {

// a. List Interface (ArrayList)

List<String> list = new ArrayList<>(Arrays.asList("Apple", "Banana", "Cherry"));

System.out.println("List: " + list);

list.remove("Apple");

System.out.println("After removing 'Apple': " + list);

// b. Set Interface (HashSet)

Set<String> set = new HashSet<>(Arrays.asList("Apple", "Banana", "Cherry"));

System.out.println("\nSet: " + set);

set.remove("Apple");

System.out.println("After removing 'Apple': " + set);

// c. Map Interface (HashMap)

Map<String, Integer> map = new HashMap<>();

map.put("Apple", 1);

map.put("Banana", 2);

System.out.println("\nMap: " + map);

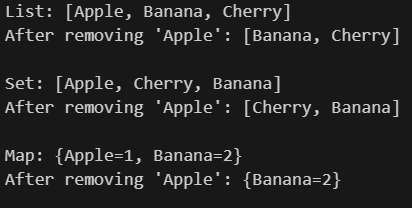
map.remove("Apple");

System.out.println("After removing 'Apple': " + map);

}

}

**Output:**

****

**Practical 5**

**Aim:** Write a program using various swing components to design Java applications to accept a student resume.

**Program:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class StudentResumeForm extends JFrame implements ActionListener {

private JTextField nameField, contactField, educationField, skillsField;

private JTextArea addressArea;

private JButton submitButton;

public StudentResumeForm() {

setTitle("Student Resume Form");

setSize(400,400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new GridLayout(6,2,10,10));

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

nameField = new JTextField();

JLabel contactLabel = new JLabel("Contact:");

contactField = new JTextField();

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

addressArea = new JTextArea(3,20);

JScrollPane addressScrollPane = new JScrollPane(addressArea);

JLabel educationLabel = new JLabel("Educational Background:");

educationField = new JTextField();

JLabel skillsLabel = new JLabel("Skills:");

skillsField = new JTextField();

submitButton = new JButton("Submit");

submitButton.addActionListener(this);

add(nameLabel);

add(nameField);

add(contactLabel);

add(contactField);

add(addressLabel);

add(addressScrollPane);

add(educationLabel);

add(educationField);

add(skillsLabel);

add(skillsField);

add(new JLabel()); // Empty Label for layout

add(submitButton);

setVisible(true);

}

@Override

public void actionPerformed(ActionEvent e) {

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

String name = nameField.getText();

String contact = contactField.getText();

String address = addressArea.getText();

String education = educationField.getText();

String skills = skillsField.getText();

JOptionPane.showMessageDialog(this,

"Resume Submitted:\n" +

"Name: " + name + "\n" +

"Contact: " + contact + "\n" +

"Address: " + address + "\n" +

"Educational Background: " + education + "\n" +

"Skills: " + skills,

"Resume Information",

JOptionPane.INFORMATION\_MESSAGE);

}

}

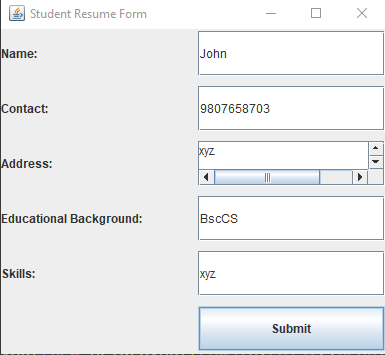
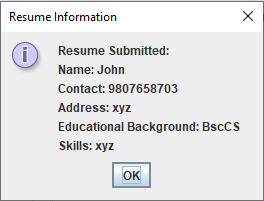
public static void main(String[] args) {

new StudentResumeForm();

}

}

**Output:**

** **

**Practical 6**

**Aim:** a. Write a JDBC program that displays the data of a given table.

b. Write a JDBC program to return the data of a specific record from a given table.

c. Write a JDBC program to insert/update/delete records into a given table.

**Program (a):**

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class DisplayTableData {

public static void main(String[] args) {

String jdbcUrl = "jdbc:mysql://localhost:3306/your\_database\_name";

String username = "your\_username";

String password = "your\_password";

String query = "SELECT \* FROM students";

try (Connection connection = DriverManager.getConnection(jdbcUrl, username, password);

Statement statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery(query)) {

while (resultSet.next()) {

int id = resultSet.getInt("id");

String name = resultSet.getString("name");

int age = resultSet.getInt("age");

System.out.println("ID: " + id + ", Name: " + name + ", Age: " + age);

}

} catch (SQLException e) {

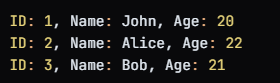
System.err.println("Error occurred while executing the query: " + e.getMessage());

}

}

}

**Output:**



**Program (b):**

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.Scanner;

public class RetrieveSpecificRecord {

public static void main(String[] args) {

String jdbcUrl = "jdbc:mysql://localhost:3306/your\_database\_name";

String username = "your\_username";

String password = "your\_password";

String query = "SELECT \* FROM students WHERE id = ?";

try (Scanner scanner = new Scanner(System.in)) {

System.out.print("Enter the ID of the record you want to retrieve: ");

int recordId = scanner.nextInt();

Connection connection = DriverManager.getConnection(jdbcUrl, username, password);

PreparedStatement preparedStatement = connection.prepareStatement(query);

preparedStatement.setInt(1, recordId);

ResultSet resultSet = preparedStatement.executeQuery();

if (resultSet.next()) {

int id = resultSet.getInt("id");

String name = resultSet.getString("name");

int age = resultSet.getInt("age");

System.out.println("Record Found:");

System.out.println("ID: " + id + ", Name: " + name + ", Age: " + age);

} else {

System.out.println("No record found with ID: " + recordId);

}

resultSet.close();

preparedStatement.close();

connection.close();

} catch (Exception e) {

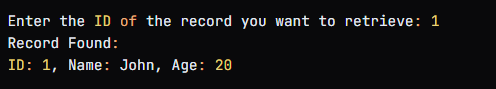
e.printStackTrace();

}

}

}

**Output:**



**Program (c):**

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.util.Scanner;

public class JDBCOperations {

public static void main(String[] args) {

String jdbcUrl = "jdbc:mysql://localhost:3306/your\_database\_name";

String username = "your\_username";

String password = "your\_password";

try (Scanner scanner = new Scanner(System.in)) {

Connection connection = DriverManager.getConnection(jdbcUrl, username, password);

System.out.println("Select an operation:");

System.out.println("1. Insert a record");

System.out.println("2. Update a record");

System.out.println("3. Delete a record");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

insertRecord(scanner, connection);

break;

case 2:

updateRecord(scanner, connection);

break;

case 3:

deleteRecord(scanner, connection);

break;

default:

System.out.println("Invalid choice. Please select 1, 2, or 3.");

}

connection.close();

} catch (Exception e) {

e.printStackTrace();

}

}

private static void insertRecord(Scanner scanner, Connection connection) throws Exception {

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

int id = scanner.nextInt();

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

String name = scanner.next();

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

int age = scanner.nextInt();

String insertQuery = "INSERT INTO students (id, name, age) VALUES (?, ?, ?)";

try (PreparedStatement preparedStatement = connection.prepareStatement(insertQuery)) {

preparedStatement.setInt(1, id);

preparedStatement.setString(2, name);

preparedStatement.setInt(3, age);

int rowsAffected = preparedStatement.executeUpdate();

if (rowsAffected > 0) {

System.out.println("Record inserted successfully.");

} else {

System.out.println("Failed to insert record.");

}

}

}

private static void updateRecord(Scanner scanner, Connection connection) throws Exception {

System.out.print("Enter ID of the record to update: ");

int id = scanner.nextInt();

System.out.print("Enter new Name: ");

String name = scanner.next();

System.out.print("Enter new Age: ");

int age = scanner.nextInt();

String updateQuery = "UPDATE students SET name = ?, age = ? WHERE id = ?";

try (PreparedStatement preparedStatement = connection.prepareStatement(updateQuery)) {

preparedStatement.setString(1, name);

preparedStatement.setInt(2, age);

preparedStatement.setInt(3, id);

int rowsAffected = preparedStatement.executeUpdate();

if (rowsAffected > 0) {

System.out.println("Record updated successfully.");

} else {

System.out.println("Failed to update record.");

}

}

}

private static void deleteRecord(Scanner scanner, Connection connection) throws Exception {

System.out.print("Enter ID of the record to delete: ");

int id = scanner.nextInt();

String deleteQuery = "DELETE FROM students WHERE id = ?";

try (PreparedStatement preparedStatement = connection.prepareStatement(deleteQuery)) {

preparedStatement.setInt(1, id);

int rowsAffected = preparedStatement.executeUpdate();

if (rowsAffected > 0) {

System.out.println("Record deleted successfully.");

} else {

System.out.println("Failed to delete record.");

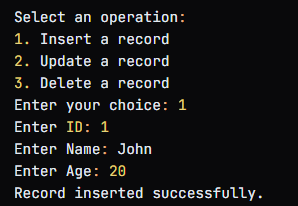
}

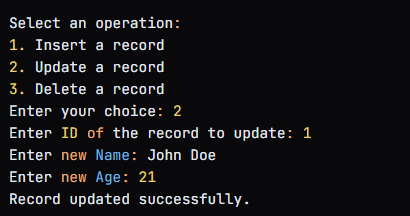
}

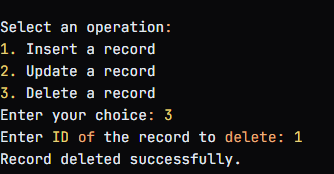
}

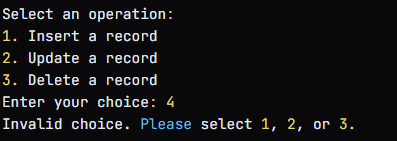
}

**Output:**









**Practical 7**

**Aim:** a) Construct a simple calculator using the Java Swing with Minimum functionality b) Construct a GUI using Java Swings to accepts details of a record of a given table and submit it to the database using JDBC technology on the click of a button

**Program (a):**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class SimpleCalculator extends JFrame implements ActionListener {

private JTextField display;

private JButton[] numberButtons;

private JButton addButton, subButton, mulButton, divButton, eqButton, clrButton;

private JPanel panel;

private double num1 = 0, num2 = 0, result = 0;

private char operator;

public SimpleCalculator() {

setTitle("Simple Calculator");

setSize(400, 500);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

display = new JTextField();

display.setFont(new Font("Arial", Font.PLAIN, 36));

display.setEditable(false);

display.setHorizontalAlignment(SwingConstants.RIGHT);

numberButtons = new JButton[10];

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

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

numberButtons[i].setFont(new Font("Arial", Font.PLAIN, 28));

numberButtons[i].addActionListener(this);

}

addButton = new JButton("+");

subButton = new JButton("-");

mulButton = new JButton("\*");

divButton = new JButton("/");

eqButton = new JButton("=");

clrButton = new JButton("C");

addButton.setFont(new Font("Arial", Font.PLAIN, 28));

subButton.setFont(new Font("Arial", Font.PLAIN, 28));

mulButton.setFont(new Font("Arial", Font.PLAIN, 28));

divButton.setFont(new Font("Arial", Font.PLAIN, 28));

eqButton.setFont(new Font("Arial", Font.PLAIN, 28));

clrButton.setFont(new Font("Arial", Font.PLAIN, 28));

addButton.addActionListener(this);

subButton.addActionListener(this);

mulButton.addActionListener(this);

divButton.addActionListener(this);

eqButton.addActionListener(this);

clrButton.addActionListener(this);

panel = new JPanel();

panel.setLayout(new GridLayout(4, 4, 10, 10));

panel.add(numberButtons[1]);

panel.add(numberButtons[2]);

panel.add(numberButtons[3]);

panel.add(addButton);

panel.add(numberButtons[4]);

panel.add(numberButtons[5]);

panel.add(numberButtons[6]);

panel.add(subButton);

panel.add(numberButtons[7]);

panel.add(numberButtons[8]);

panel.add(numberButtons[9]);

panel.add(mulButton);

panel.add(clrButton);

panel.add(numberButtons[0]);

panel.add(eqButton);

panel.add(divButton);

add(display, BorderLayout.NORTH);

add(panel);

setVisible(true);

}

@Override

public void actionPerformed(ActionEvent e) {

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

if (e.getSource() == numberButtons[i]) {

display.setText(display.getText().concat(String.valueOf(i)));

}

}

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

num1 = Double.parseDouble(display.getText());

operator = '+';

display.setText("");

}

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

num1 = Double.parseDouble(display.getText());

operator = '-';

display.setText("");

}

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

num1 = Double.parseDouble(display.getText());

operator = '\*';

display.setText("");

}

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

num1 = Double.parseDouble(display.getText());

operator = '/';

display.setText("");

}

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

num2 = Double.parseDouble(display.getText());

switch (operator) {

case '+':

result = num1 + num2;

break;

case '-':

result = num1 - num2;

break;

case '\*':

result = num1 \* num2;

break;

case '/':

result = num1 / num2;

break;

}

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

num1 = result;

}

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

display.setText("");

}

}

public static void main(String[] args) {

new SimpleCalculator();

}

}

**Output:**

****

**Program (b):**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

public class RecordEntryForm extends JFrame implements ActionListener {

private JTextField idField, nameField, ageField;

private JButton submitButton;

private final String jdbcUrl = "jdbc:mysql://localhost:3306/your\_database\_name";

private final String username = "your\_username";

private final String password = "your\_password";

public RecordEntryForm() {

setTitle("Record Entry Form");

setSize(400, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

setLayout(new GridLayout(4, 2, 10, 10));

JLabel idLabel = new JLabel("ID:");

idField = new JTextField();

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

nameField = new JTextField();

JLabel ageLabel = new JLabel("Age:");

ageField = new JTextField();

submitButton = new JButton("Submit");

submitButton.addActionListener(this);

add(idLabel);

add(idField);

add(nameLabel);

add(nameField);

add(ageLabel);

add(ageField);

add(submitButton);

setVisible(true);

}

public void actionPerformed(ActionEvent e) {

int id = Integer.parseInt(idField.getText());

String name = nameField.getText();

int age = Integer.parseInt(ageField.getText());

try {

Connection connection = DriverManager.getConnection(jdbcUrl, username, password);

String insertQuery = "INSERT INTO students (id, name, age) VALUES (?, ?, ?)";

PreparedStatement preparedStatement = connection.prepareStatement(insertQuery);

preparedStatement.setInt(1, id);

preparedStatement.setString(2, name);

preparedStatement.setInt(3, age);

int rowsAffected = preparedStatement.executeUpdate();

if (rowsAffected > 0) {

JOptionPane.showMessageDialog(this, "Record inserted successfully.");

} else {

JOptionPane.showMessageDialog(this, "Failed to insert record.");

}

preparedStatement.close();

connection.close();

} catch (Exception ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage());

}

}

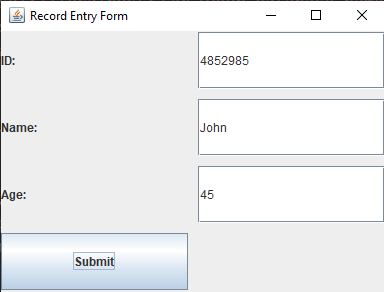
public static void main(String[] args) {

new RecordEntryForm();

}

}

**Output:**

****

**Practical 8**

**Aim:** a. Write a Servlet that accepts a username from a HTML form and stores it as a cookie. Write another Servlet that returns the value of this cookie and displays it.

b. Write a Servlet that displays the name and values of the cookies stored on the client.

c. Write a Servlet that accepts a Username from a HTML from and stores it as a session variable. Write another Servlet that returns the value of this session variable and display it.

**Program (a):**

Code:

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<title>Set Username</title>

</head>

<body>

<h2>Enter Your Username</h2>

<form action="setCookie.jsp" method="post">

<label for="username">Username:</label>

<input type="text" id="username" name="username" required>

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

</form>

</body>

</html>

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ page import="javax.servlet.http.Cookie" %>

<%@ page import="javax.servlet.http.HttpServletRequest" %>

<%@ page import="javax.servlet.http.HttpServletResponse" %>

<%

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

Cookie cookie = new Cookie("username", username);

cookie.setMaxAge(60 \* 60 \* 24);

response.addCookie(cookie);

response.sendRedirect("getCookie.jsp");

%>

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ page import="javax.servlet.http.Cookie" %>

<%@ page import="javax.servlet.http.HttpServletRequest" %>

<%

String username = null;

Cookie[] cookies = request.getCookies();

if (cookies != null) {

for (Cookie cookie : cookies) {

if (cookie.getName().equals("username")) {

username = cookie.getValue();

break;

}

}

}

%>

<!DOCTYPE html>

<html>

<head>

<title>Display Cookie</title>

</head>

<body>

<h2>Cookie Value</h2>

<%

if (username != null) {

%>

<p>Welcome, <strong><%= username %></strong>!</p>

<%

} else {

%>

<p>No username found in cookie.</p>

<%

}

%>

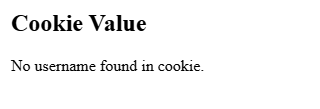
</body>

</html>

**Output:**



IF THERE IS NO COOKIE



**Program (b):**

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ page import="javax.servlet.http.Cookie" %>

<%@ page import="javax.servlet.http.HttpServletRequest" %>

<%

// Retrieve cookies from the request

Cookie[] cookies = request.getCookies();

%>

<!DOCTYPE html>

<html>

<head>

<title>Display Cookies</title>

</head>

<body>

<h2>Stored Cookies</h2>

<table border="1">

<tr>

<th>Cookie Name</th>

<th>Cookie Value</th>

</tr>

<%

if (cookies != null) {

for (Cookie cookie : cookies) {

%>

<tr>

<td><%= cookie.getName() %></td>

<td><%= cookie.getValue() %></td>

</tr>

<%

}

} else {

%>

<tr>

<td colspan="2">No cookies found.</td>

</tr>

<%

}

%>

</table>

</body>

</html>

<%

Cookie cookie1 = new Cookie("username", "JohnDoe");

cookie1.setMaxAge(60 \* 60 \* 24); // 1 day

response.addCookie(cookie1);

Cookie cookie2 = new Cookie("email", "john@example.com");

cookie2.setMaxAge(60 \* 60 \* 24); // 1 day

response.addCookie(cookie2);

%>

<html>

<body>

<h2>Cookies Set</h2>

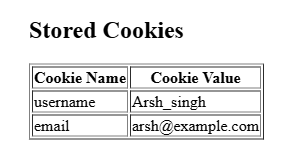
<p>Two cookies have been set: username and email.</p>

<a href="displayCookies.jsp">View Cookies</a>

</body>

</html>

**Output:**

****

**If no cookie are there**

****

**Program (c):**

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<title>Set Username</title>

</head>

<body>

<h2>Enter your Username</h2>

<form action="storeUsername.jsp" method="post">

<label for="username">Username:</label>

<input type="text" id="username" name="username" required>

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

</form>

</body>

</html>

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ page import="javax.servlet.http.HttpSession" %>

<%

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

HttpSession session = request.getSession();

session.setAttribute("username", username);

response.sendRedirect("getUsername.jsp");

%>

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%

// Retrieve the username from the session

HttpSession session = request.getSession();

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

%>

<!DOCTYPE html>

<html>

<head>

<title>Welcome</title>

</head>

<body>

<h2>Welcome</h2>

<%

if (username != null) {

%>

<h1>Hello, <%= username %>!</h1>

<%

} else {

%>

<h1>No username found in session.</h1>

<%

}

%>

</body>

</html>

**Output:**

****

**Practical 9**

**Aim:** a. Write a registration servlet that accepts the data for a given table and stores it in the database. b. Write a Servlet that displays all the records of a table.

**Program (a):**

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL UNIQUE,

password VARCHAR(50) NOT NULL,

email VARCHAR(100) NOT NULL

);

<!DOCTYPE html>

<html>

<head>

<title>Registration Form</title>

</head>

<body>

<h2>User Registration</h2>

<form action="registerUser .jsp" method="post">

<label for="username">Username:</label>

<input type="text" id="username" name="username" required>

<br><br>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required>

<br><br>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required>

<br><br>

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

</form>

</body>

</html>

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

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

<%

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

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

String email = request.getParameter("email");

String dbURL = "jdbc:mysql://localhost:3306/database";

String dbUser = "username";

String dbPassword = "password";

Connection conn = null;

PreparedStatement pstmt = null;

try {

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

conn = DriverManager.getConnection(dbURL, dbUser , dbPassword);

String sql = "INSERT INTO users (username, password, email) VALUES (?, ?, ?)";

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, username);

pstmt.setString(2, password);

pstmt.setString(3, email);

int rowsAffected = pstmt.executeUpdate();

if (rowsAffected > 0) {

%>

<h2>Registration successful! Welcome, <%= username %>!</h2>

<%

} else {

%>

<h2>Registration failed. Please try again.</h2>

<%

}

} catch (Exception e) {

e.printStackTrace();

%>

<h2>Error: <%= e.getMessage() %></h2>

<%

} finally {

try {

if (pstmt != null) pstmt.close();

if (conn != null) conn.close();

} catch (SQLException e) {

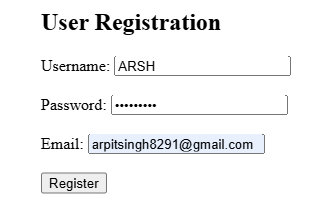
e.printStackTrace();

}

}

%>

**Output:**



**Program (b):**

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

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

<!DOCTYPE html>

<html>

<head>

<title>User Records</title>

</head>

<body>

<h2>User Records</h2>

<table border="1">

<tr>

<th>ID</th>

<th>Username</th>

<th>Password</th>

<th>Email</th>

</tr>

<%

String dbURL = "jdbc:mysql://localhost:3306/your\_database

String dbUser = "username";

String dbPassword = "password";

Connection conn = null;

Statement stmt = null;

ResultSet rs = null;

try {

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

conn = DriverManager.getConnection(dbURL, dbUser , dbPassword);

stmt = conn.createStatement

String sql = "SELECT \* FROM users";

rs = stmt.executeQuery(sql);

while (rs.next()) {

int id = rs.getInt("id");

String username = rs.getString("username");

String password = rs.getString("password");

String email = rs.getString("email");

%>

<tr>

<td><%= id %></td>

<td><%= username %></td>

<td><%= password %></td>

<td><%= email %></td>

</tr>

<%

}

} catch (Exception e) {

e.printStackTrace();

%>

<tr>

<td colspan="4">Error: <%= e.getMessage() %></td>

</tr>

<%

} finally {

try {

if (rs != null) rs.close();

if (stmt != null) stmt.close();

if (conn != null) conn.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

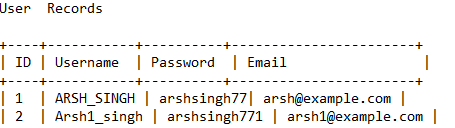
%>

</table>

</body>

</html>

**Output:**

****

**Practical 10**

**Aim:** a. Write a JSP that accepts a username from a HTML form and stores it as a cookie. Write another JSP that returns the value of this cookie and displays it.

b. Write a JSP that displays the name and values of the cookies stored on the client.

c. Write a JSP that accepts a Username from a HTML from and stores it as a session variable. Write another JSP that returns the value of this session variable and displays it.

**Program (a):**

<!DOCTYPE html>

<html>

<head>

<title>Username Form</title>

</head>

<body>

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

<label for="username">Enter your username:</label>

<input type="text" id="username" name="username" required>

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

</form>

</body>

</html>

Set username servlet:

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/setUsername")

public class SetUsernameServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

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

Cookie cookie = new Cookie("username", username);

cookie.setMaxAge(60 \* 60 \* 24);

response.addCookie(cookie);

response.sendRedirect("getUsername");

}

}

Get username servlet

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/getUsername")

public class GetUsernameServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

Cookie[] cookies = request.getCookies();

String username = null;

if (cookies != null) {

for (Cookie cookie : cookies) {

if ("username".equals(cookie.getName())) {

username = cookie.getValue();

break;

}

}

}

response.setContentType("text/html");

response.getWriter().println("<html><body>");

if (username != null) {

response.getWriter().println("<h1>Welcome, " + username + "!</h1>");

} else {

response.getWriter().println("<h1>No username found in cookies.</h1>");

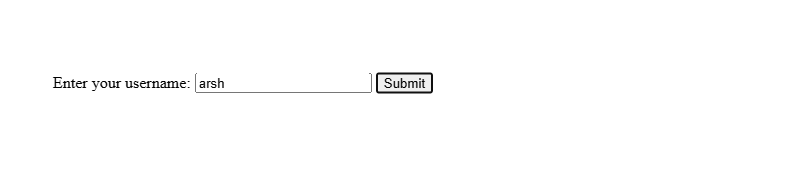
}

response.getWriter().println("</body></html>");

}

}

**Output:**

****

****

**Program (b):**

<html>

<body>

<h1>Stored Cookies</h1>

<table border='1'>

<tr>

<th>Cookie Name</th>

<th>Cookie Value</th>

</tr>

<tr>

<td>username</td>

<td>arsh</td>

</tr>

<tr>

<td>sessionID</td>

<td>abc123</td>

</tr>

</table>

</body>

</html>

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/displayCookies")

public class CookieDisplayServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

// Get the cookies from the request

Cookie[] cookies = request.getCookies();

response.setContentType("text/html");

response.getWriter().println("<html><body>");

response.getWriter().println("<h1>Stored Cookies</h1>");

if (cookies != null && cookies.length > 0) {

response.getWriter().println("<table border='1'><tr><th>Cookie Name</th><th>Cookie Value</th></tr>");

for (Cookie cookie : cookies) {

response.getWriter().println("<tr><td>" + cookie.getName() + "</td><td>" + cookie.getValue() + "</td></tr>");

}

response.getWriter().println("</table>");

} else {

response.getWriter().println("<p>No cookies found.</p>");

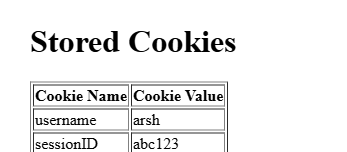
}

response.getWriter().println("</body></html>");

}

}

**Output:**

****

**Program (c):**

<!DOCTYPE html>

<html>

<head>

<title>Username Form</title>

</head>

<body>

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

<label for="username">Enter your username:</label>

<input type="text" id="username" name="username" required>

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

</form>

</body>

</html>

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

@WebServlet("/setUsernameSession")

public class SetUsernameSessionServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

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

HttpSession session = request.getSession();

if (username != null && !username.trim().isEmpty()) {

session.setAttribute("username", username.trim());

}

response.sendRedirect("getUsernameSession");

}

}

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

@WebServlet("/getUsernameSession")

public class GetUsernameSessionServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

// Retrieve the session

HttpSession session = request.getSession();

// Get the username from the session

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

response.setContentType("text/html");

response.getWriter().println("<html><body>");

if (username != null) {

response.getWriter().println("<h1>Welcome, " + username + "!</h1>");

} else {

response.getWriter().println("<h1>No username found in session.</h1>");

}

response.getWriter().println("</body></html>");

}

}

**Output:**





**Practical 11**

**Aim:** a. Write a JSP code that accepts username and password from HTML file and validates the user from the database.

b. Write a registration JSP that accepts the data for a given table and stores it in the database.

c. Write a JSP that displays all the records of a table.

**Program (a):**

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL,

password VARCHAR(50) NOT NULL

);

INSERT INTO users (username, password) VALUES ('testuser', 'testpassword');

<!DOCTYPE html>

<html>

<head>

<title>Login Form</title>

</head>

<body>

<h2>Login</h2>

<form action="validateUser .jsp" method="post">

<label for="username">Username:</label>

<input type="text" id="username" name="username" required>

<br><br>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required>

<br><br>

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

</form>

</body>

</html>

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

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

<%

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

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

// Database connection variables

String dbURL = "jdbc:mysql://localhost:3306/database";

String dbUser = "username";

String dbPassword = "password";

Connection conn = null;

PreparedStatement pstmt = null;

ResultSet rs = null;

try {

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

conn = DriverManager.getConnection(dbURL, dbUser , dbPassword);

String sql = "SELECT \* FROM users WHERE username = ? AND password = ?";

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, username);

pstmt.setString(2, password);

// Execute the query

rs = pstmt.executeQuery();

// Check if user exists

if (rs.next()) {

%>

<h2>Welcome, <%= username %>!</h2>

<%

} else {

%>

<h2>Invalid username or password.</h2>

<%

}

} catch (Exception e) {

e.printStackTrace();

%>

<h2>Error: <%= e.getMessage() %></h2>

<%

} finally {

// Close resources

try {

if (rs != null) rs.close();

if (pstmt != null) pstmt.close();

if (conn != null) conn.close();

} catch (SQLException e) {

e.printStackTrace();

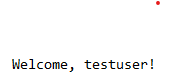
}

}

%>

**Output:**

****

****

**Program (b):**

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL,

password VARCHAR(50) NOT NULL,

email VARCHAR(100) NOT NULL

);

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

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("/register")

public class RegistrationServlet extends HttpServlet {

private static final String DB\_URL = "jdbc:mysql://localhost:3306/database\_name";

private static final String DB\_USER = "username

private static final String DB\_PASSWORD = "password";

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

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

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

String email = request.getParameter("email");

// Validate input

if (username == null || password == null || email == null ||

username.isEmpty() || password.isEmpty() || email.isEmpty()) {

response.getWriter().println("<h1>Error: All fields are required!</h1>");

return;

}

try (Connection connection = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD)) {

String sql = "INSERT INTO users (username, password, email) VALUES (?, ?, ?)";

try (PreparedStatement statement = connection.prepareStatement(sql)) {

statement.setString(1, username);

statement.setString(2, password);

statement.setString(3, email);

int rowsInserted = statement.executeUpdate();

if (rowsInserted > 0) {

response.getWriter().println("<h1>Registration successful!</h1>");

} else {

response.getWriter().println("<h1>Error: Registration failed!</h1>");

}

}

} catch (Exception e) {

e.printStackTrace();

response.getWriter().println("<h1>Error: " + e.getMessage() + "</h1>");

}

}

}

<!DOCTYPE html>

<html>

<head>

<title>Registration Form</title>

</head>

<body>

<h2>Register</h2>

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

<label for="username">Username:</label>

<input type="text" id="username" name="username" required><br><br>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required><br><br>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required><br><br>

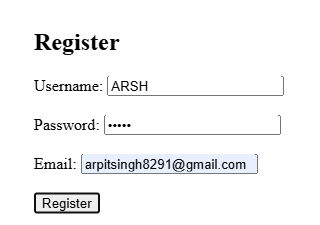
<input type="submit" value="Register">

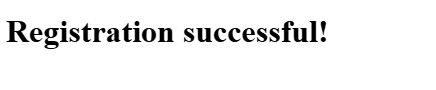
</form>

</body>

</html>

**Output:**





**Program (c):**

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL,

password VARCHAR(50) NOT NULL,

email VARCHAR(100) NOT NULL

);

INSERT INTO users (username, password, email) VALUES ('arsh\_singh', 'password123', 'arsh@example.com');

INSERT INTO users (username, password, email) VALUES ('arsh1\_singh', 'password456', 'arsh1@example.com');

<html>

<body>

<h1>User Records</h1>

<table border='1'>

<tr>

<th>ID</th>

<th>Username</th>

<th>Email</th>

</tr>

<tr>

<td>1</td>

<td>arsh\_singh</td>

<td>arsh@example.com</td>

</tr>

<tr>

<td>2</td>

<td>arsh1\_singh</td>

<td>arsh1@example.com</td>

</tr>

</table>

</body>

</html>

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

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("/displayRecords")

public class DisplayRecordsServlet extends HttpServlet {

private static final String DB\_URL = "jdbc:mysql://localhost:3306/database\_name";

private static final String DB\_USER = "username";

private static final String DB\_PASSWORD = "password";

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

StringBuilder htmlResponse = new StringBuilder();

htmlResponse.append("<html><body>");

htmlResponse.append("<h1>User Records</h1>");

htmlResponse.append("<table border='1'><tr><th>ID</th><th>Username</th><th>Email</th></tr>");

try (Connection connection = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD)) {

String sql = "SELECT id, username, email FROM users";

try (PreparedStatement statement = connection.prepareStatement(sql);

ResultSet resultSet = statement.executeQuery()) {

while (resultSet.next()) {

int id = resultSet.getInt("id");

String username = resultSet.getString("username");

String email = resultSet.getString("email");

htmlResponse.append("<tr>");

htmlResponse.append("<td>").append(id).append("</td>");

htmlResponse.append("<td>").append(username).append("</td>");

htmlResponse.append("<td>").append(email).append("</td>");

htmlResponse.append("</tr>");

}

}

} catch (Exception e) {

e.printStackTrace();

htmlResponse.append("<tr><td colspan='3'>Error: ").append(e.getMessage()).append("</td></tr>");

}

htmlResponse.append("</table>");

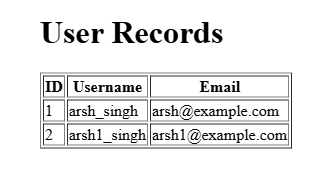
htmlResponse.append("</body></html>");

response.getWriter().println(htmlResponse.toString());

}

}

**Output:**

****

**Practical 12**

**Aim:** Write a Java application to encoding and decoding JSON in Java

**Program:**

import com.google.gson.Gson;

class Employee {

    String name;

    int age;

    String[] skills;

    Employee(String name, int age, String[] skills) {

        this.name = name;

        this.age = age;

        this.skills = skills;

    }

}

public class JsonExample {

    public static void main(String[] args) {

        Gson gson = new Gson();

        Employee emp = new Employee("Alice", 30, new String[]{"Java", "Python", "JavaScript"});

        String json = gson.toJson(emp);

        System.out.println("JSON Representation: " + json);

        Employee empFromJson = gson.fromJson(json, Employee.class);

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

    }

}

**Output:**

****