Lab 6

WAP to create a package that access the number of external class as well as samepackage.

```
package myPackage;
public class MyPackageClass { public void samePackageMethod() {
System.out.println("Method in the same package class");
}
}
package OtherPackage; public class ExternalClass {
public void externalMethod() { System.out.println("Method in an external class");
}
}
import myPackage.MyPackageClass; import OtherPackage.ExternalClass; public class
AccessClassesDemo { public static void main(String[] args) {
MyPackageClass myPackageObj = new MyPackageClass();myPackageObj.samePackageMethod();
ExternalClass externalObj = new ExternalClass();externalObj.externalMethod();
}
}
```

WAP that import user defined package and access the member variable of classesthat contained by package.

```
public class MyClass1{ public int variable1=42;}package myPackage; public class MyClass2{
public String variable2=”Hello, World”;
}
Import myPackage.MyClass1;Import myPackage.MyClass2;
Public class ImportUserDefinedPackage{Public static void main(String[] args){ MyClass1
obj1=new MyClass1(); MyClass2 obj 2=new MyClass2();
System.out.println(“Member variable in MyClass1:’’+obj.variable1); System.out.println(“Member
variable in MyClass2:”+obj2.variable2);
}
}
```

Lab 9

WAP to connected the java with mysql Server.

```
import java.sql.Connection; import java.sql.DriverManager;import java.sql.ResultSet; import
java.sql.Statement; public class sql_test {
public static void main(String args[]) {try
{
Class.forName("com.mysql,jdbc,Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/student", "root" ,"");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("select * from second"); while(rs.next())
System.out.println(rs.getInt(1) +" " + rs.getString(2) + " " + rs.getString(3));con.close();
}
catch (Exception e)
{
System.out.println(e);
}
}
}
```

Lab 10

WAP to create Frame the display the student information and store data in MySQLserver. Perform the below operation like store,update, delete, clear.

```
package registration_form.registration_form;class App
{
public static void main( String[] args )
{
new RegistrationForm();
}
}
package registration_form.registration_form;import javax.swing.*;
import java.awt.event.*;import java.awt.*; import java.sql.*;
public class RegistrationForm implements ActionListener {JFrame frame;
String[] gender = { "Male", "Female" }; JLabel nameLabel = new JLabel("NAME");
JLabel genderLabel = new JLabel("GENDER");
JLabel fatherNameLabel = new JLabel("FATHER NAME");JLabel passwordLabel = new
JLabel("PASSWORD");
JLabel confirmPasswordLabel = new JLabel("CONFIRM PASSWORD");JLabel cityLabel = new
JLabel("CITY");
JLabel emailLabel = new JLabel("EMAIL"); JTextField nameTextField = new JTextField();
JComboBox genderComboBox = new JComboBox(gender);JTextField fatherTextField = new
JTextField(); JPasswordField passwordField = new JPasswordField();
JPasswordField confirmPasswordField = new JPasswordField();JTextField cityTextField = new
JTextField();
JTextField emailTextField = new JTextField(); JButton registerButton = new
JButton("REGISTER");JButton resetButton = new JButton("RESET"); JButton updateButton = new
JButton("UPDATE"); JButton deleteButton = new JButton("DELETE"); JPanel recordsPanel;
JTextArea recordsTextArea;
// Constructor RegistrationForm() { createWindow(); setLocationAndSize();
addComponentsToFrame();createRecordsPanel(); actionEvent();
}
public void createWindow() {frame = new JFrame();
frame.setTitle("Registration Form"); frame.setBounds(40, 40, 380, 600);
frame.getContentPane().setBackground(Color.CYAN); frame.getContentPane().setLayout(null);
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); frame.setResizable(false);
}
```

```
public void setLocationAndSize() { nameLabel.setBounds(20, 20, 40, 70);
genderLabel.setBounds(20, 70, 80, 70);
fatherNameLabel.setBounds(20, 120, 100, 70);
passwordLabel.setBounds(20, 170, 100, 70);
confirmPasswordLabel.setBounds(20, 220, 140, 70);
cityLabel.setBounds(20, 270, 100, 70);
emailLabel.setBounds(20, 320, 100, 70);
nameTextField.setBounds(180, 43, 165, 23);
genderComboBox.setBounds(180, 93, 165, 23);
fatherTextField.setBounds(180, 143, 165, 23);
passwordField.setBounds(180, 193, 165, 23);
confirmPasswordField.setBounds(180, 243, 165, 23);
cityTextField.setBounds(180, 293, 165, 23);
emailTextField.setBounds(180, 343, 165, 23);
registerButton.setBounds(70, 400, 100, 35);
resetButton.setBounds(220, 400, 100, 35);
updateButton.setBounds(70, 450, 100, 35);
deleteButton.setBounds(220, 450, 100, 35);
}
public void addComponentsToFrame() {frame.add(nameLabel); frame.add(genderLabel);
frame.add(fatherNameLabel); frame.add(passwordLabel); frame.add(confirmPasswordLabel);
frame.add(cityLabel); frame.add(emailLabel); frame.add(nameTextField);
frame.add(genderComboBox); frame.add(fatherTextField); frame.add(passwordField);
frame.add(confirmPasswordField); frame.add(cityTextField); frame.add(emailTextField);
frame.add(registerButton); frame.add(resetButton); frame.add(updateButton);
frame.add(deleteButton); }
public void actionEvent() { registerButton.addActionListener(this);
resetButton.addActionListener(this); updateButton.addActionListener(this);
deleteButton.addActionListener(this);
}
public void actionPerformed(ActionEvent e) {if (e.getSource() == registerButton) {
try {
// Creating Connection Object
Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/myDatabase",
"root","root");
// Prepared Statement
PreparedStatement Pstatement = connection.prepareStatement("insert into student
values(?,?,?,?,?,?,?)");
// Specifying the values of its parameter Pstatement.setString(1, nameTextField.getText());
Pstatement.setString(2, genderComboBox.getSelectedItem().toString());Pstatement.setString(3,
fatherTextField.getText()); Pstatement.setString(4, passwordField.getText()); Pstatement.setString(5,
confirmPasswordField.getText()); Pstatement.setString(6, cityTextField.getText());
```

```
Pstatement.setString(7, emailTextField.getText());
// Checking for the Password match
if (passwordField.getText().equalsIgnoreCase(confirmPasswordField.getText())) {
// Executing query Pstatement.executeUpdate();
JOptionPane.showMessageDialog(null, "Data Registered Successfully");
} else {
JOptionPane.showMessageDialog(null, "Password did not match");
}
} catch (SQLException e1) {e1.printStackTrace();
}
} else if (e.getSource() == resetButton) {
// Clearing Fields nameTextField.setText(""); genderComboBox.setSelectedItem("Male");
fatherTextField.setText(""); passwordField.setText(""); confirmPasswordField.setText("");
cityTextField.setText(""); emailTextField.setText("");
} else if (e.getSource() == updateButton) {try {
// Creating Connection Object
Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/myDatabase",
"root","root");
// Prepared Statement for updating a student's information based on email (primary key)
String updateQuery = "UPDATE student SET gender=?, fname=?, passwd=?, conpasswd=?, city=?
WHEREemail=?";
PreparedStatement Pstatement = connection.prepareStatement(updateQuery);
// Specifying the values of its parameters
Pstatement.setString(1, genderComboBox.getSelectedItem().toString());Pstatement.setString(2,
fatherTextField.getText()); Pstatement.setString(3, passwordField.getText()); Pstatement.setString(4,
cityTextField.getText()); Pstatement.setString(5, confirmPasswordField.getText());
Pstatement.setString(6, emailTextField.getText()); // Use email as the primary keyint rowsUpdated =
Pstatement.executeUpdate();
if (rowsUpdated > 0) {
JOptionPane.showMessageDialog(null, "Data Updated Successfully");
} else {
JOptionPane.showMessageDialog(null, "No student with this email found.");
}
} catch (SQLException e1) {e1.printStackTrace();
}
} else if (e.getSource() == deleteButton) {try {
// Creating Connection Object
Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/myDatabase",
"root","root");
// Prepared Statement for deleting a student's information based on email (primary key)String
deleteQuery = "DELETE FROM student WHERE email=?";
PreparedStatement Pstatement = connection.prepareStatement(deleteQuery);
// Specifying the values of its parameters
Pstatement.setString(1, emailTextField.getText()); // Use email as the primary keyint rowsDeleted =
Pstatement.executeUpdate();
if (rowsDeleted > 0) {
JOptionPane.showMessageDialog(null, "Data Deleted Successfully");
// Clear the form after deletion
```

```
nameTextField.setText(""); genderComboBox.setSelectedItem("Male");fatherTextField.setText("");
passwordField.setText(""); confirmPasswordField.setText(""); cityTextField.setText("");
emailTextField.setText("");
} else {
JOptionPane.showMessageDialog(null, "No student with this email found.");
}
} catch (SQLException e1) {e1.printStackTrace();
}
}
}
public void createRecordsPanel() { recordsPanel = new JPanel(); recordsPanel.setBounds(400, 20,
300, 500);
recordsPanel.setBorder(BorderFactory.createTitledBorder("Database Records"));recordsTextArea =
new JTextArea(20, 25);
recordsTextArea.setEditable(false);
JScrollPane scrollPane = new JScrollPane(recordsTextArea);recordsPanel.add(scrollPane);
frame.add(recordsPanel);
}
// Method to fetch and display records from the databasepublic void displayRecords() {
try {
// Creating Connection Object
Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/myDatabase",
"root","root");
// Create a Statement to execute SQL queries Statement statement = connection.createStatement();
// Execute a SELECT query to fetch records String selectQuery = "SELECT * FROM student";
ResultSet resultSet = statement.executeQuery(selectQuery);
// Display records in the recordsTextArea StringBuilder records = new StringBuilder();while
(resultSet.next()) {
String name = resultSet.getString("name"); String gender = resultSet.getString("gender");
String fatherName = resultSet.getString("fname");String city = resultSet.getString("city");
String email = resultSet.getString("email"); records.append("Name: ").append(name).append("\n");
records.append("Gender: ").append(gender).append("\n");
records.append("Father Name: ").append(fatherName).append("\n");records.append("City:
").append(city).append("\n"); records.append("Email: ").append(email).append("\n\n");
}
recordsTextArea.setText(records.toString());
// Close the resourcesresultSet.close(); statement.close(); connection.close();
} catch (SQLException e) {e.printStackTrace();
}
}
public static void main(String[] args) }
```

```
new RegistrationForm();
}
}
```