TRIBHUVAN UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY

Lab Report on

Advanced Java Programming

Submitted To

Er. Jeewan Rai

Department of Computer Science and Information Technology

Nagarjuna College of IT

Submitted By

Bibek Angdembe (23957/076)

Jan 2024

Table of contents

Er. Jeewan Rai

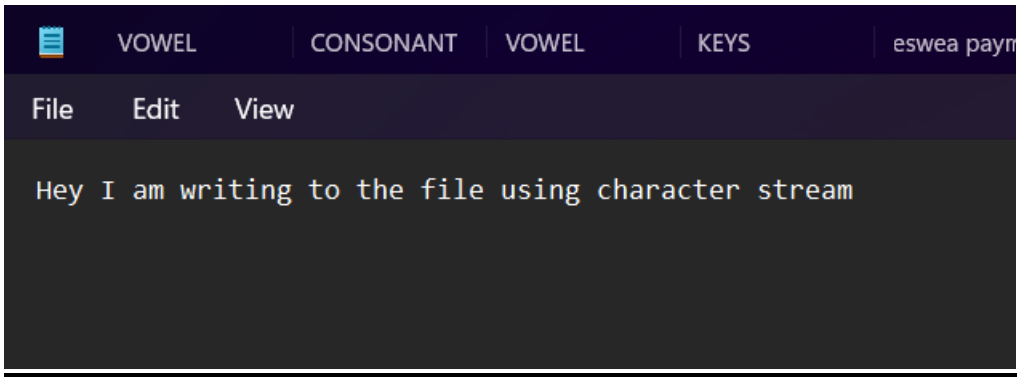
1) Write a simple java program to read from and write to files.

Source Code

```
import java.io.*;
public class WriteReadExample {
   public static void main(String[] args){
      try{
         //wrtiting to file
         FileWriter file = new FileWriter("testFile.txt");
         file.write("Hey I am writing to the file using character stream");
          file.close();
         //reading from a file
         FileReader fileRead = new FileReader("testFile.txt");
         int i;
         while((i= fileRead.read()) != -1){
            System.out.print((char)i);
         }

         fileRead.close();
      }catch(IOException e){
         System.out.println(e.getMessage());
      }
   }
}
```

Output

```
--- exec:3.1.0:exec (default-cli) @ AssignmentOne ---
Hey I am writing to the file using character stream
------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------
Total time:  3.248 s
Finished at: 2024-02-20T09:29:08+05:45
```



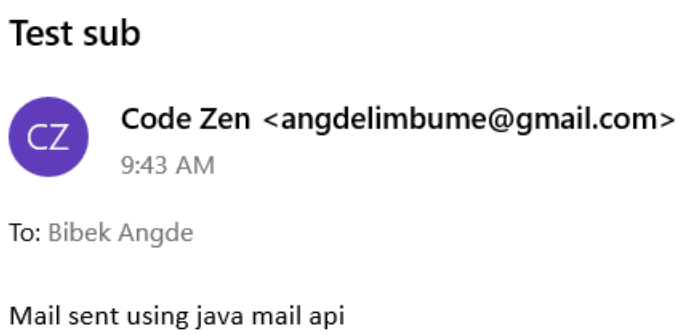
2) Write a java program for sending email messages

Source Code

```
import java.util.*;
import javax.mail.*;
import javax.mail.internet.*;
import javax.activation.*;
import javax.mail.Session;
import javax.mail.Transport;



public class SendEmail
{


  public static void main(String [] args)
  {
    String recipient = "Bibekangdembay@gmail.com";
     String sender = "Angdelimbume@gmail.com";
    String host = "127.0.0.1";
    Properties properties = System.getProperties();
      properties.setProperty("mail.smtp.host", host);
      Session session = Session.getDefaultInstance(properties);
    try
    {
      // MimeMessage object.
      MimeMessage message = new MimeMessage(session);
        message.setFrom(new InternetAddress(sender));
      message.addRecipient(Message.RecipientType.TO,                    new
InternetAddress(recipient));
        message.setSubject("Test sub");
```

```
            message.setText("Mail sent using java mail api");
        Transport.send(message);
    }
    catch (MessagingException mex)
    {
      mex.printStackTrace();
    }
  }
}
```

Output



3) Write a program using swing components to find simple interest. Use text fields for inputs and output. Your program should display the result when the user presses a button.

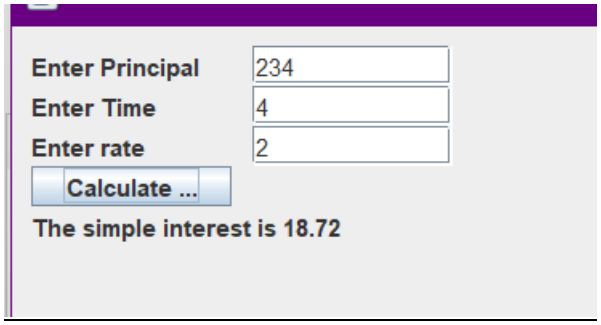
Source Code

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import static javax.swing.WindowConstants.EXIT_ON_CLOSE;
public class SimpleInterest {
   private JLabel p , t ,r ,result;
   private JTextField pf,tf,rf;
   private JButton b;

   public SimpleInterest(){
      JFrame f = new JFrame();
      f.setDefaultCloseOperation(EXIT_ON_CLOSE);
      p = new JLabel("Enter Principal");
      p.setBounds(10,10,100,20);
      f.add(p);

      pf = new JTextField();
      pf.setBounds(120,10,100,20);
      f.add(pf);

       t = new JLabel("Enter Time");
      t.setBounds(10,30,100,20);
      f.add(t);

      tf = new JTextField();
```

```
tf.setBounds(120,30,100,20);
f.add(tf);

 r = new JLabel("Enter rate");
r.setBounds(10,50,100,20);
f.add(r);

rf = new JTextField();
rf.setBounds(120,50,100,20);
f.add(rf);

b = new JButton("Calculate SI");
b.setBounds(10,70,100,20);
f.add(b);

result = new JLabel();
result.setBounds(10,90,400,20);
f.add(result);

b.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        double prin = Double.parseDouble(pf.getText());
        double time = Double.parseDouble(tf.getText());
        double rate = Double.parseDouble(rf.getText());

        double SI = (prin*time*rate)/100;
        result.setText("The simple interest is " + SI);
    }
});
f.setLayout(null);
```

```
        f.setSize(500,500);
        f.setVisible(true);

    }
    public static void main(String[] args) {
        new SimpleInterest();
    }
}
```

Output



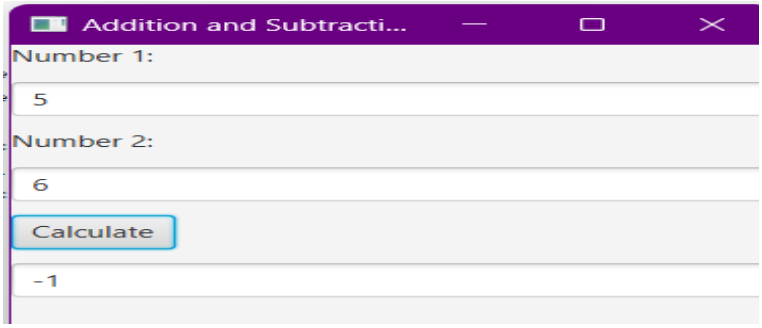
4) Write a program using JavaFX components to add and subtract of two numbers. Use text fields for inputs and output. Your program should display the addition result when the user presses a button and subtract when release the button.

Source Code

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
public class AddSub extends Application {
  private TextField num1Field, num2Field, resultField;
   @Override
   public void start(Stage primaryStage) {
      Label num1Label = new Label("Number 1:");
      num1Field = new TextField();
      Label num2Label = new Label("Number 2:");
      num2Field = new TextField();
      Button calculateButton = new Button("Calculate");
      calculateButton.setOnMousePressed(e -> handleAddition());
      calculateButton.setOnMouseReleased(e -> handleSubtraction());
      resultField = new TextField();
      resultField.setEditable(false);
      VBox root = new VBox(10);
      root.getChildren().addAll(num1Label, num1Field, num2Label, num2Field, calculateButton, resultField);
      Scene scene = new Scene(root, 300, 250);
      primaryStage.setTitle("Addition and Subtraction");
      primaryStage.setScene(scene);
```

```
        primaryStage.show();
    }
    private void handleAddition() {
        try {
            int num1 = Integer.parseInt(num1Field.getText());
            int num2 = Integer.parseInt(num2Field.getText());
            int result = num1 + num2;
            resultField.setText(Integer.toString(result));
        } catch (NumberFormatException e) {
            resultField.setText("Invalid input");
        }   }
    private void handleSubtraction() {
        try {
            int num1 = Integer.parseInt(num1Field.getText());
            int num2 = Integer.parseInt(num2Field.getText());
            int result = num1 - num2;
            resultField.setText(Integer.toString(result));
        } catch (NumberFormatException e) {
            resultField.setText("Invalid input");
        } }

    public static void main(String[] args) {
        launch(args);
    }}
```

Output



5) Write a Java programs using RMI to find the addition of two numbers.

Step 1: Create the remote interface

```
//Adder.java
import java.rmi.*;
public interface Adder extends Remote {
   public int Add(int a , int b) throws RemoteException;
}
```

Step 2: Implementation of the remote interface

```
//AddImplement.java
import java.rmi.*;
import java.rmi.server.*;
public  class AddImplement  extends UnicastRemoteObject implements Adder{
   AddImplement() throws RemoteException{
      super();
   }
   @Override
   public int Add(int a , int b) throws RemoteException{
      return a+b;
   }
}
```

Step 3: Compile the implementation class and create the stub and skeleton objects using the rmic tool

```
rmic AdderRemote
```

Step 4: Start the registry service by rmiregistry tool

rmiregistry portNumber

Step 5: Create and start the remote application

```
//ClientRmi.java
import java.rmi.*;
public class ClientRmi {
   public static void main(String[] args){
      try{
         Adder stub =(Adder)Naming.lookup("rmi://localhost:5000/test");
         int sum = stub.Add(5,6);
         System.out.println("Return sum " + sum);
      }catch(Exception e){
         System.out.println(e);
      }
   }
}
```

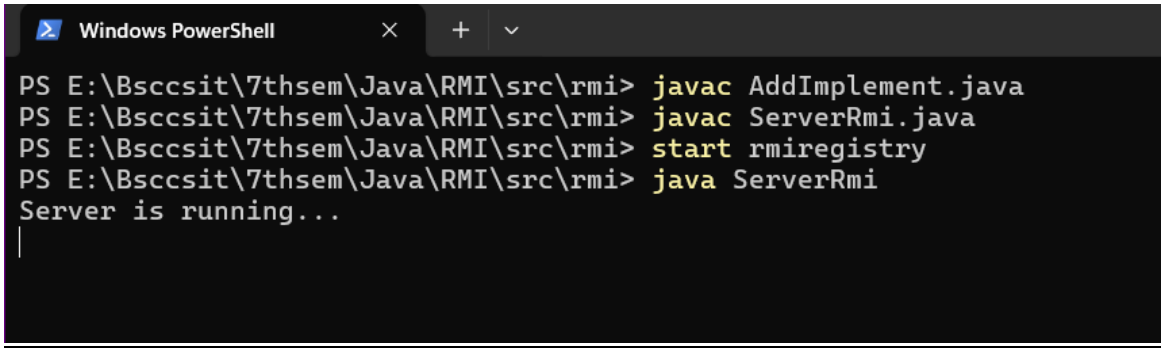
Step 6: Create and start the client application

```
//ServerRmi.java
import java.rmi.*;
public class ServerRmi {
   public static void main(String[] args){
      try{
         Adder skeleton =  (Adder) new  AddImplement();
```

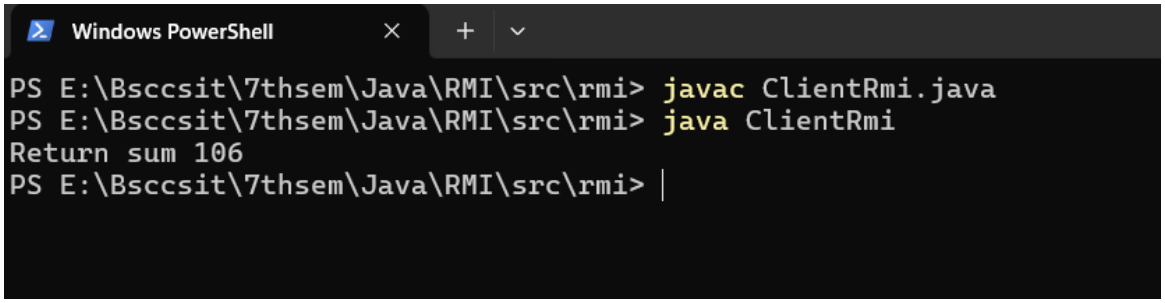
```
        Naming.rebind("rmi://localhost:5000/test", skeleton);
        }
        catch(Exception e){
            System.out.println(e);
        }
    }
}
```

Output

//Server



//Client



6) Create a simple Servlet that reads and displays data from HTML form.

Assume a HTML form with two fields username and password.

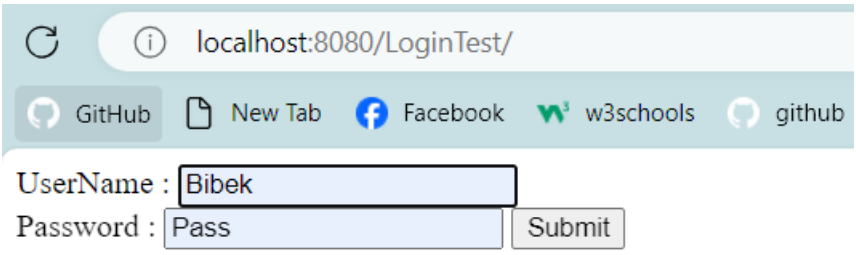
Source Code

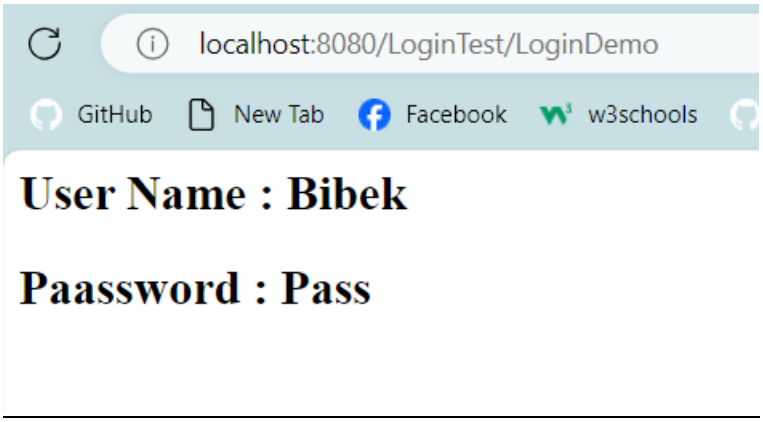
```
//index.html
<!DOCTYPE html>
<html>
   <head>
      <title>TODO supply a title</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>
      <form action="LoginDemo" method="post" >
         UserName :
         <input type="text" name="user_name" />
         </br>
         Password :
          <input type="text" name="password" />
          <input type="submit" value="Submit" />
      </form>
   </body>
</html>


//web.xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
   <servlet>
      <servlet-name>LoginDemo</servlet-name>
```

```
        <servlet-class>LoginDemo</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>LoginDemo</servlet-name>
        <url-pattern>/LoginDemo</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
</web-app>

//LoginDemo.java

import java.io.*;
import javax.servlet.http.*;

public class LoginDemo extends HttpServlet {

   @Override
    protected void doPost(HttpServletRequest request,HttpServletResponse response){
        response.setContentType("text/html;charset=utf-8");


        try{
            String name = request.getParameter("user_name");
        String pass = request.getParameter("password");
            PrintWriter out = response.getWriter();
        out.println("<h2>User Name : " + name + "</h2>");
```

```
        out.println("<h2>Paassword : " + pass + "</h2>");
        }catch(Exception e){
            System.out.println(e);
        }
    }
}
```

Output





---

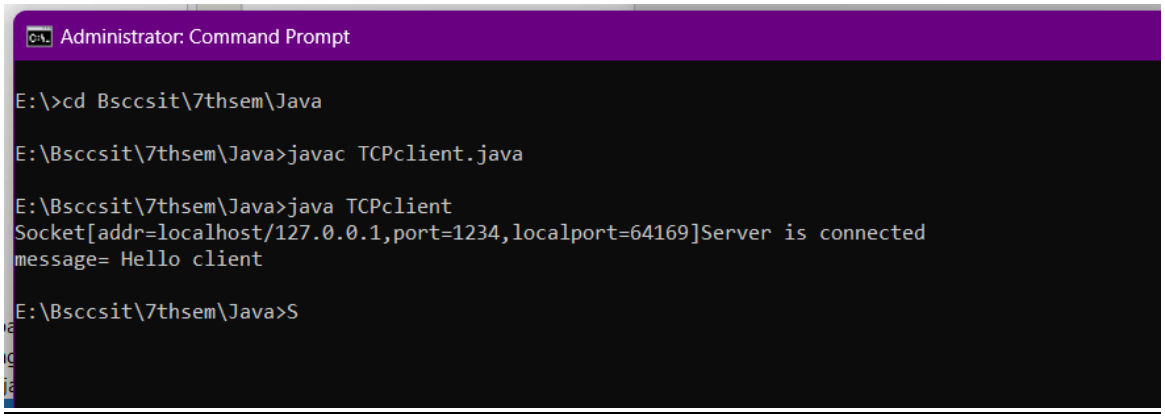
7) Write TCP and UDP based socket programming using Java to communicate between client and server.

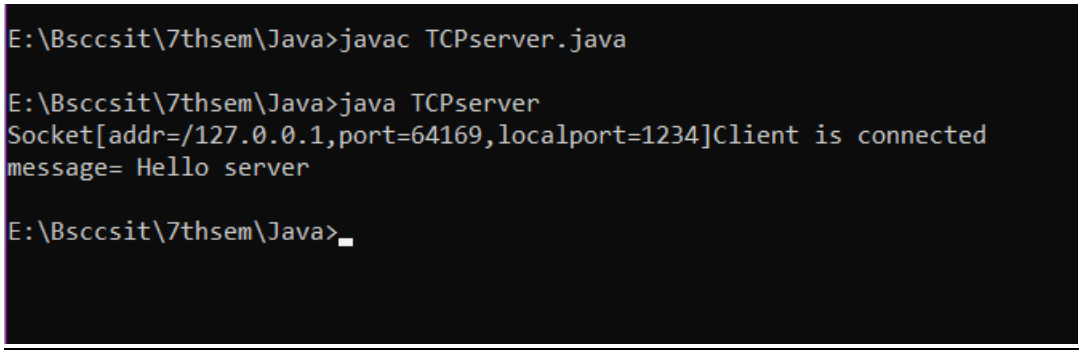
Source Code

```
//TCPclient.java
import java.io.*;
import java.net.*;
public class TCPclient {
   public static void main(String[] args){
      try{
         Socket socket = new Socket("localhost",1234);
         System.out.println(socket + "Server is connected");
         DataOutputStream out=new DataOutputStream(socket.getOutputStream());
        DataInputStream in=new DataInputStream(socket.getInputStream());
        out.writeUTF("Hello server");
        String str=(String) in.readUTF();
        System.out.println("message= "+str);
        socket.close();
      }catch(Exception e){
         System.out.println(e);
      }
   }}

//TCPserver.java
import java.io.*;
import java.net.*;
public class TCPserver {
   public static void main(String[] args){
      try{
```

```
 ServerSocket server=new ServerSocket(1234);
        Socket client=server.accept();//establishes connection
        System.out.println(client + "Client is connected");
        DataOutputStream out=new DataOutputStream(client.getOutputStream());
        DataInputStream in=new DataInputStream(client.getInputStream());
        String str=(String) in.readUTF();
        System.out.println("message= "+str);
        out.writeUTF("Hello client");
        client.close();
}catch(Exception e)
{
   System.out.println(e);}
} }
```

Output





8) Write a Java programs to insert, select, delete and update the students records (fullname, email, phoneno, gender, title) in database using JDBC

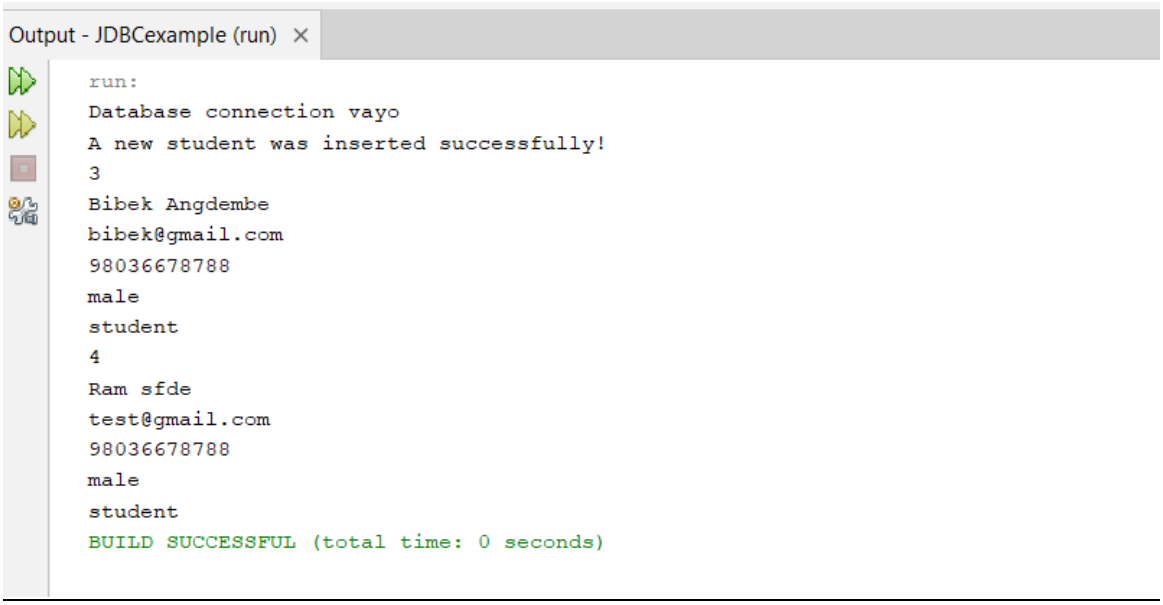
Source Code

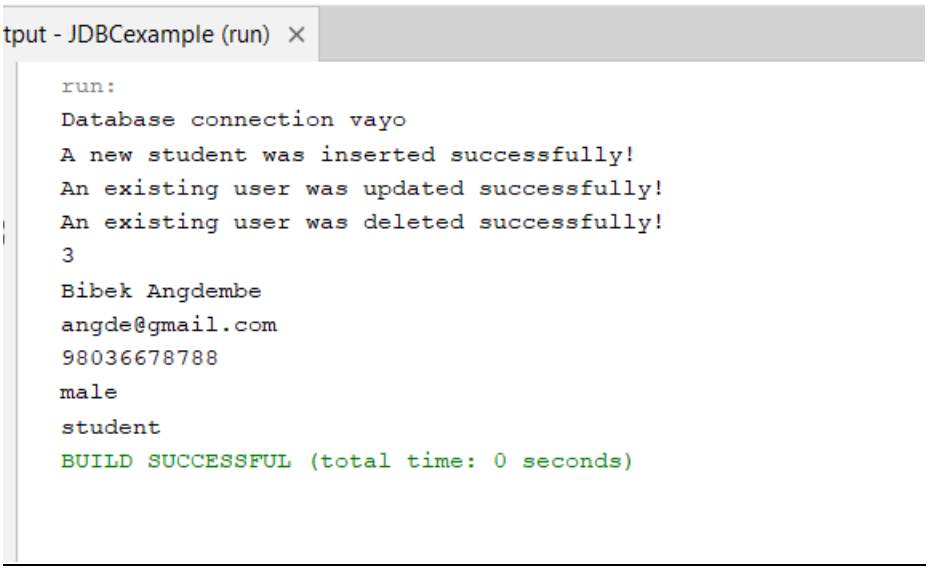
```
package jdbcexample;

import java.sql.*;
public class CrudExmp {

   static String driver = "com.mysql.cj.jdbc.Driver";
   static String url = "jdbc:mysql://localhost:3306/";
   static String db = "college_javaad";
   static String user = "root";
   static String pw = "";

   public static void main(String[] args) {
      Connection conn = null;
      Statement st = null;
      ResultSet set = null;

      try {
         conn = DriverManager.getConnection(url + db, user, pw);
         if (conn != null) {
            System.out.println("Database connection vayo");

            String sqlQ = "INSERT INTO users ( fullname, email,phoneno,gender,title) VALUES (?, ?, ?, ?, ?)";

            PreparedStatement statement = conn.prepareStatement(sqlQ);
            statement.setString(1, "Ram sfde");
            statement.setString(2, "test@gmail.com");
```

```
statement.setString(3, "98036678788");
statement.setString(4, "male");
statement.setString(5, "student");

int rowsInserted = statement.executeUpdate();
if (rowsInserted > 0) {
   System.out.println("A new student was inserted successfully!");
}


String Updatesql = "UPDATE users SET email = ? WHERE fullname=?";

PreparedStatement state = conn.prepareStatement(Updatesql);
state.setString(1, "angde@gmail.com");
state.setString(2, "Bibek Angdembe");

int rowsUpdated = state.executeUpdate();
if (rowsUpdated > 0) {
   System.out.println("An existing user was updated successfully!");
}

String delsql = "DELETE FROM users WHERE fullname = ?";

PreparedStatement statem = conn.prepareStatement(delsql);
statem.setString(1, "Ram sfde");

int rowsDeleted = statem.executeUpdate();
if (rowsDeleted > 0) {
   System.out.println("An existing user was deleted successfully!");
}
```

```
String sql = "SELECT * FROM users";
            st = conn.createStatement();
            set = st.executeQuery(sql);
            while (set.next()) {
               System.out.println(set.getString("id"));
               System.out.println(set.getString("fullname"));
               System.out.println(set.getString("email"));
               System.out.println(set.getString("phoneno"));
               System.out.println(set.getString("gender"));
               System.out.println(set.getString("title"));
            }

         }
         }catch (SQLException e) {
         System.out.println(e);
      }
      }

  }
```

Output


Output - JDBCexample (run)
run:
Database connection vayo
A new student was inserted successfully!
3
Bibek Angdembe
bibek@gmail.com
98036678788
male
student
4
Ram sfde
test@gmail.com
98036678788
male
student
BUILD SUCCESSFUL (total time: 0 seconds)


tput - JDBCexample (run)
run:
Database connection vayo
A new student was inserted successfully!
An existing user was updated successfully!
An existing user was deleted successfully!
3
Bibek Angdembe
angde@gmail.com
98036678788
male
student
BUILD SUCCESSFUL (total time: 0 seconds)