Java Assignment 1: MODULE 1

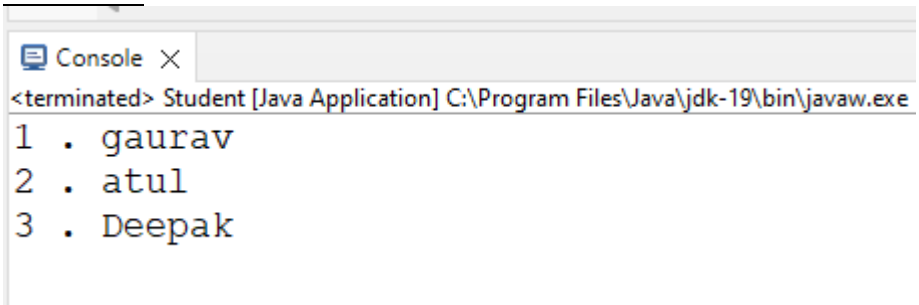
Name – CHINMAYA GARNAIK
Class - FYMCA(B)
PRN – 1132220942

Q1) Simple Java Program using JDBC Driver.

CODE: →

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
public class Student {
public static void main(String[] args)throws Exception {
Class.forName("com.mysql.cj.jdbc.Driver");
Connection connection=
DriverManager.getConnection("jdbc:mysql://localhost:3306/db1","root"
,"mysql");
Statement statement= connection.createStatement();
ResultSet resultSet= statement.executeQuery("select * from
customer");
while(resultSet.next()) {
int iid= resultSet.getInt("id");
String Fame= resultSet.getString("Name");
System.out.println(iid+" . "+ Fame);
}
}
}
```

OUTPUT:



Q2) Simple Java program explaining Statement, Prepared Statement & Callable Statement.

Code: →

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.Statement;
import java.sql.ResultSet;
import java.util.Scanner;

public class jdbctest2 {
    public static void main(String args[]) throws Exception {
        Scanner sc = new Scanner(System.in);
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/db1", "root",
"mysql");
        Statement statement = connection.createStatement();
        ResultSet resultSet = statement.executeQuery("select * from
customer");
        while (resultSet.next()) {
            int iid = resultSet.getInt("id");
            String tame = resultSet.getString("Name");
            System.out.println(iid + " " + tame);

        }
        System.out.println(
                "There are two ages in the table customer 12 and 18 . Please
enter the age to know the customer names: ");
        int a = sc.nextInt();
        PreparedStatement ps = connection.prepareStatement("select name from
db1.customer where age=?");
        ps.setInt(1, a);
        ResultSet rsp = ps.executeQuery();
        while (rsp.next()) {
            System.out.println("Name " + rsp.getString(1));
        }

    }
}
```

Output:

```
1 gaurav
2 atul
3 Deepak
There are two ages in the table customer 12 and 18 . Please enter the age to know the customer names:
18
Name gaurav
Name Deepak
PS D:\MIT\Semester 2\java practicall\jdbc\jdbcvs>
```

Q.3) Simple Java program explaining JDBC with AWT.

CODE: →

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class jdbctest3 extends Frame implements ActionListener {
    private Label lblName;
    private Label lblRollNo;
    private TextField txtName;
    private TextField txtRollNo;
    private Button btnSubmit;
    private Button btnReset;
    private Connection con;
    private Statement stmt;
    private ResultSet rs;

    public jdbctest3() {
        lblName = new Label("Name:");
        lblRollNo = new Label("Roll No:");
        txtName = new TextField(20);
        txtRollNo = new TextField(20);
        btnSubmit = new Button("Submit");
        btnReset = new Button("Reset");

        setLayout(new FlowLayout());
        add(lblName);
        add(txtName);
        add(lblRollNo);
        add(txtRollNo);
        add(btnSubmit);
        add(btnReset);

        btnSubmit.addActionListener(this);
        btnReset.addActionListener(this);
        setTitle("JDBC Example");
        setSize(250, 150);
        setVisible(true);

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/db1", "root",
"mysql");
            stmt = con.createStatement();
        } catch (Exception e) {
            System.out.println("Error: " + e);
        }
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == btnSubmit) {
            try {
                String name = txtName.getText();
                int rollNo = Integer.parseInt(txtRollNo.getText());
```

```
                String sql = "INSERT INTO student (name, roll_no) VALUES ('" +
name + "', '" + rollNo + "')";
                stmt.executeUpdate(sql);
                System.out.println("Record inserted successfully");
            } catch (Exception ex) {
                System.out.println("Error: " + ex);
            }
        } else {
            txtName.setText("");
            txtRollNo.setText("");
        }
    }

    public static void main(String[] args) {
        jdbctest3 example = new jdbctest3();
    }
}
```

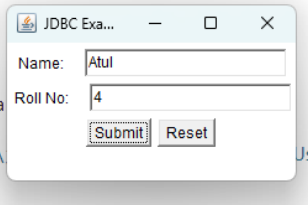
OUTPUT:

```
PROBLEMS 3   OUTPUT   DEBUG CONSOLE   TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka

PS D:\MIT\Semester 2\java practicall\jdbc\jdbcvs>  & 'C:\Program Files\Java\                    Users
st3'
Record inserted successfully
```



Q4} Create and interface for students Educational data and stored it in database.

Code: →

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class jdbctest4 extends Frame implements ActionListener {
    private TextField tfName = new TextField();
    private TextField tfRollNo = new TextField();
    private TextField tfMarks = new TextField();
    private Button btnSave = new Button("Save");

    public jdbctest4() {
        setLayout(new FlowLayout());
        add(new Label("Name:"));
        add(tfName);
        add(new Label("Roll No:"));
        add(tfRollNo);
        add(new Label("Marks:"));
        add(tfMarks);
        add(btnSave);
        btnSave.addActionListener(this);
        setSize(400, 200);
        setVisible(true);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == btnSave) {
            try {
                // Load the driver
                Class.forName("com.mysql.cj.jdbc.Driver");

                // Connect to the database
                Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/db2", "root",
"mysql");

                PreparedStatement pstmt = conn
                        .prepareStatement("INSERT INTO Students (name, rollno,
marks) VALUES (?, ?, ?)");
                pstmt.setString(1, tfName.getText());
                pstmt.setInt(2, Integer.parseInt(tfRollNo.getText()));
                pstmt.setInt(3, Integer.parseInt(tfMarks.getText()));

                int result = pstmt.executeUpdate();

                if (result == 1) {
                    tfName.setText("");
```

```
                    tfRollNo.setText("");
                    tfMarks.setText("");
                    System.out.println("Data saved successfully.");
                } else {
                    System.out.println("Error saving data.");
                }

                conn.close();
            } catch (Exception e) {
                System.out.println("Error: " + e.getMessage());
            }
        }
    }

    public static void main(String[] args) {
        new jdbctest4();
    }
}
```

Q5) Create Library database and perform add, update, delete, show operations. Create Interface for it and perform database transactions on It.

Code: →

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class LibraryGUI extends Frame implements ActionListener {
    private TextField tfBookId = new TextField();
    private TextField tfBookName = new TextField();
    private TextField tfAuthor = new TextField();
    private Button btnAdd = new Button("Add");
    private Button btnUpdate = new Button("Update");
    private Button btnDelete = new Button("Delete");
    private Button btnShow = new Button("Show");

    public LibraryGUI() {
        setLayout(new FlowLayout());
        add(new Label("Book ID:"));
        add(tfBookId);
        add(new Label("Book Name:"));
        add(tfBookName);
        add(new Label("Author:"));
        add(tfAuthor);
        add(btnAdd);
        add(btnUpdate);
        add(btnDelete);
        add(btnShow);
        btnAdd.addActionListener(this);
        btnUpdate.addActionListener(this);
        btnDelete.addActionListener(this);
        btnShow.addActionListener(this);
        setSize(400, 200);
        setVisible(true);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });
    }

   public void actionPerformed(ActionEvent ae) {
       if (ae.getSource() == btnAdd) {
           try {

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


               Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/test", "user",
"password");
```

```
                PreparedStatement pstmt = conn.prepareStatement("INSERT INTO
Library (bookid, bookname, author) VALUES (?, ?, ?)");
                pstmt.setInt(1, Integer.parseInt(tfBookId.getText()));
                pstmt.setString(2, tfBookName.getText());
                pstmt.setString(3, tfAuthor.getText());

                int result = pstmt.executeUpdate();

                if (result == 1) {
                    tfBookId.setText("");
                    tfBookName.setText("");
                    tfAuthor.setText("");
                    System.out.println("Data saved successfully.");
                } else {
                    System.out.println("Error saving data.");
                }


                conn.close();
            } catch (Exception e) {
                System.out.println("Error: " + e.getMessage());
            }
        } else if (ae.getSource() == btnUpdate) {
            try {

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


                Connection conn;

    Driver DB() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/db_name", "username",
"password");

            PreparedStatement pstmt = conn
                    .prepareStatement("INSERT INTO Students (name, rollno,
marks) VALUES (?, ?, ?)");
            pstmt.setString(1, tfName.getText());
            pstmt.setInt(2, Integer.parseInt(tfRollNo.getText()));
            pstmt.setInt(3, Integer.parseInt(tfMarks.getText()));

            int result = pstmt.executeUpdate();

            if (result == 1) {
                tfName.setText("");
                tfRollNo.setText("");
                tfMarks.setText("");
                System.out.println("Data saved successfully.");
            } else {
                System.out.println("Error saving data.");
            }
```

```
            conn.close();
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

    }

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

Q6. Create JDBC Application for table

Student (Roll, Name, Email, Address, Course_id)
Course(Courseid, Course_name, Fees)
Result(Roll, Percentage)
i) Design a good interface.
ii) Perform New, Save, Edit, Delete operations on table.
iii) Use Try Catch Finally and Transaction.

CODE: →

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class jdbctest6 extends Frame implements ActionListener {

    Label lblRoll, lblName, lblEmail, lblAddress, lblCourseID, lblResult,
lblPercentage;
    TextField tfRoll, tfName, tfEmail, tfAddress, tfCourseID, tfPercentage;
    Button btnNew, btnSave, btnEdit, btnDelete;

    Connection con;
    Statement stmt;

    jdbctest6() {
        setLayout(new FlowLayout());
        lblRoll = new Label("Roll");
        lblName = new Label("Name");
        lblEmail = new Label("Email");
        lblAddress = new Label("Address");
        lblCourseID = new Label("Course ID");
        lblResult = new Label("Result");
        lblPercentage = new Label("Percentage");
        tfRoll = new TextField(20);
        tfName = new TextField(20);
        tfEmail = new TextField(20);
        tfAddress = new TextField(20);
        tfCourseID = new TextField(20);
        tfPercentage = new TextField(20);
        btnNew = new Button("New");
        btnSave = new Button("Save");
        btnEdit = new Button("Edit");
        btnDelete = new Button("Delete");

        add(lblRoll);
        add(tfRoll);
        add(lblName);
        add(tfName);
        add(lblEmail);
        add(tfEmail);
        add(lblAddress);
        add(tfAddress);
```

```
        add(lblCourseID);
        add(tfCourseID);
        add(lblResult);
        add(lblPercentage);
        add(tfPercentage);
        add(btnNew);
        add(btnSave);
        add(btnEdit);
        add(btnDelete);

        btnNew.addActionListener(this);
        btnSave.addActionListener(this);
        btnEdit.addActionListener(this);
        btnDelete.addActionListener(this);

        setTitle("JDBC AWT Application");
        setSize(400, 300);
        setVisible(true);
        connectToDB();
    }

    public void actionPerformed(ActionEvent ae) {
        String str = ae.getActionCommand();

        if (str.equals("New")) {
            clearFields();
        }

        else if (str.equals("Save")) {
            try {

                int roll = Integer.parseInt(tfRoll.getText());
                String name = tfName.getText();
                String email = tfEmail.getText();
                String address = tfAddress.getText();

                String query = "INSERT INTO table_name(id, name, email,
address) VALUES(" + id + ", '" + name + "', '"
                        + email + "', '" + address + "')";
                stmt.executeUpdate(query);
            } catch (Exception e) {
                System.out.println(e);
            }
        } else if (str.equals("Edit")) {
            try {
                int id = Integer.parseInt(tfId.getText());
                String name = tfName.getText();
                String email = tfEmail.getText();
                String address = tfAddress.getText();

                String query = "UPDATE table_name SET name='" + name + "',
email='" + email + "', address='" + address
                        + "' WHERE id=" + id;
                stmt.executeUpdate(query);
            } catch (Exception e) {
                System.out.println(e);
```

```
            }
        }

        else if (str.equals("Delete")) {
            try {

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

                String query = "DELETE FROM table_name WHERE id=" + id;
                stmt.executeUpdate(query);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }

    void connectToDB() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/db4", "root",
"mysql");

            PreparedStatement pstmt = conn
                    .prepareStatement("INSERT INTO Students (name, rollno,
marks) VALUES (?, ?, ?)");
            pstmt.setString(1, tfName.getText());
            pstmt.setInt(2, Integer.parseInt(tfRollNo.getText()));
            pstmt.setInt(3, Integer.parseInt(tfMarks.getText()));

            int result = pstmt.executeUpdate();
            if (result == 1) {
                tfName.setText("");
                tfRollNo.setText("");
                tfMarks.setText("");
                System.out.println("Data saved successfully.");
            } else {
                System.out.println("Error saving data.");
            }
            conn.close();
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

    public static void main(String[] args) {
        new jdbctest6();
    }
}
```