Institute of Computer Technology

B. Tech. Computer Science and Engineering

Semester: III

Sub: Object-Oriented Programming

Course Code: 2CSE303

Practical Number:15

Objective: To learn about the JDBC concept in java.

Problem Definition:

Q.1. What is the concept of JDBC in java? Explain with examples.

Ans:- JDBC (Java Database Connectivity) is an API to connect and execute queries with a database. It allows Java applications to interact with relational databases.

E.g:-

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;

public class JDBCBasics {
   public static void main(String[] args) {
      try {
         Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb",
"root", "password");
         Statement stmt = con.createStatement();
         System.out.println("Connection successful!");
         con.close();
      } catch (Exception e) {
         e.printStackTrace();
      }
   }
}
```

Q.2. Explain different types of drivers in Java.

Ans:- Type 1: JDBC-ODBC Bridge Driver (uses ODBC for database interaction).

Type 2: Native-API Driver (uses database vendor's client-side libraries).

Type 3: Network Protocol Driver (uses middleware for communication).

Type 4: Thin Driver (pure Java driver interacting directly with the database).

Q.3. Explain the concept of Connection class, PreparedStatement class, and Statement class and the ResultSet class in java.

Ans:- Connection: Manages the connection with the database.

Statement: Executes SQL queries.

PreparedStatement: Precompiled SQL statements for efficiency.

ResultSet: Holds data retrieved from the database.

E.g:-

```
import java.sql.*;

public class JDBCClassesExample {
   public static void main(String[] args) {
      try {
         Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb",
"root", "password");
         PreparedStatement pstmt = con.prepareStatement("SELECT *
FROM users WHERE id = ?");
```

```
            pstmt.setInt(1, 1);
            ResultSet rs = pstmt.executeQuery();
            while (rs.next()) {
               System.out.println("Name: " + rs.getString("name"));
            }
            con.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Q.4. Explain the concept of exception handling in the JDBC program.

Ans:- Exception handling in JDBC ensures smooth execution of the program in case of errors such as connection failure or invalid SQL commands.

E.g:-

```
try {
   Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb",
"root", "password");
   // SQL operations
} catch (SQLException e) {
   System.out.println("SQL Exception: " + e.getMessage());
}
```

Q.5. What is database and table? Explain create, select, insert, modify, update and delete commands.

Ans:- CREATE: CREATE TABLE Emp (emp_id INT, emp_name VARCHAR(50));

INSERT: INSERT INTO Emp VALUES (1, 'John');

SELECT: SELECT * FROM Emp;

UPDATE: UPDATE Emp SET emp_name = 'Mike' WHERE emp_id = 1;

DELETE: DELETE FROM Emp WHERE emp_id = 1;

Q.6. Design a system, where you have to create one database with the name of "Employee", and then you have to create one table with the name of "Emp" with the following column field details like (emp_id, emp_name, designation, salary, address, and contact number). After creating the database and table, you have to insert a minimum of 5 employee records in the table, and then you have to display all the table records on terminal / screen. and then you have to search individual employee records by using emp_id or by emp_name. So, for the fulfillment purpose of the above said requirement write an appropriate JDBC program in java.

Ans:- import java.sql.*;

```
public class EmployeeDatabase {
   public static void main(String[] args) {
      try {
         Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/",
"root", "jeal259372");
         Statement stmt = con.createStatement();
```

```
        // Create Database and Table
        stmt.execute("CREATE DATABASE IF NOT EXISTS Employee");
        stmt.execute("USE Employee");
        stmt.execute("CREATE TABLE IF NOT EXISTS Emp (emp_id INT
PRIMARY KEY, emp_name VARCHAR(50), designation VARCHAR(50),
salary DOUBLE, address VARCHAR(100), contact_number
VARCHAR(15))");

        // Insert Records
        stmt.execute("INSERT INTO Emp VALUES (1, 'John', 'Manager',
50000, 'NY', '1234567890')");
        stmt.execute("INSERT INTO Emp VALUES (2, 'Jane',
'Developer', 45000, 'LA', '0987654321')");

        // Display Records
        ResultSet rs = stmt.executeQuery("SELECT * FROM Emp");
        while (rs.next()) {
            System.out.println(rs.getInt("emp_id") + " " +
rs.getString("emp_name"));
        }

        // Search by ID or Name
        PreparedStatement pstmt = con.prepareStatement("SELECT *
FROM Emp WHERE emp_id = ? OR emp_name = ?");
        pstmt.setInt(1, 1);
        pstmt.setString(2, "Jane");
        ResultSet searchResult = pstmt.executeQuery();
        while (searchResult.next()) {
            System.out.println("Found: " +
searchResult.getString("emp_name"));
        }
```

```
            con.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Q.7. Write an appropriate JDBC program in java, where you have to check if the connection is created or not with the database.

Ans:-

```
import java.sql.*;
import java.util.Scanner;

public class StudentDatabase {
    public static void main(String[] args) {
        try {
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/",
"root", "password");
            Statement stmt = con.createStatement();

            // Create Database and Table
            stmt.execute("CREATE DATABASE IF NOT EXISTS Student");
            stmt.execute("USE Student");
            stmt.execute("CREATE TABLE IF NOT EXISTS Stud (stud_id INT
PRIMARY KEY, name VARCHAR(50), course VARCHAR(50), branch
VARCHAR(50), fee DOUBLE, contact_number VARCHAR(15))");

            // Insert Records
            stmt.execute("INSERT INTO Stud VALUES (1, 'Alice', 'B.Tech',
'CSE', 50000, '1234567890')");
```

```
        stmt.execute("INSERT INTO Stud VALUES (2, 'Bob', 'B.Sc', 'IT',
45000, '0987654321')");

        // Display Records
        ResultSet rs = stmt.executeQuery("SELECT * FROM Stud");
        while (rs.next()) {
            System.out.println(rs.getInt("stud_id") + " " +
rs.getString("name"));
        }

        // Search Records
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Student ID or Name:");
        String search = sc.nextLine();
        PreparedStatement pstmt = con.prepareStatement("SELECT *
FROM Stud WHERE stud_id = ? OR name = ?");
        pstmt.setString(1, search);
        pstmt.setString(2, search);
        ResultSet searchResult = pstmt.executeQuery();
        while (searchResult.next()) {
            System.out.println("Found: " +
searchResult.getString("name"));
        }

        // Modify Records
        System.out.println("Enter ID to Update Fee:");
        int id = sc.nextInt();
        System.out.println("Enter New Fee:");
        double fee = sc.nextDouble();
        pstmt = con.prepareStatement("UPDATE Stud SET fee = ?
WHERE stud_id = ?");
```

```
            pstmt.setDouble(1, fee);
            pstmt.setInt(2, id);
            pstmt.executeUpdate();
            System.out.println("Record Updated!");

            // Delete Records
            System.out.println("Enter ID to Delete:");
            int delId = sc.nextInt();
            pstmt = con.prepareStatement("DELETE FROM Stud WHERE
stud_id = ?");
            pstmt.setInt(1, delId);
            pstmt.executeUpdate();
            System.out.println("Record Deleted!");

            con.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```