ASSIGNMENT 3

NAME : VATTI DEEPAK
REG : 20BCI0276
EMAIL : vatti.deepak2020@vitstudent.ac.in

VIT VELLORE

JDBC

CODE :

```
# Following code creates the student1 table

 Create the student table if it doesn't exist
        sql = "CREATE TABLE IF NOT EXISTS student1 ("
              + "name VARCHAR(50) NOT NULL, "
              + "registernumber VARCHAR(10) NOT NULL PRIMARY KEY, "
              + "cgpa DOUBLE, "
              + "age INT, "
              + "dayscholar BOOLEAN, "
              + "hosteller BOOLEAN)";
        stmt.executeUpdate(sql);

# Following code inserts some sample data in the database

sql = "INSERT INTO student1 (name, registernumber, cgpa, age, dayscholar, hosteller) VALUES "
              + "('John Doe', '1001', 8.5, 22, true, false), "
              + "('Jane Doe', '1002', 7.2, 20, false, true), "
              + "('Bob Smith', '1003', 4.8, 19, true, true), "
              + "('Alice Jones', '1004', 6.1, 21, false, false), "
              + "('Charlie Brown', '1005', 9.0, 18, true, true)";
        stmt.executeUpdate(sql);

# Main code

package jdbc1;
```

```
import java.sql.*;
import java.sql.ResultSet;

public class JDBC1 {

   public static void main(String[] args) {

      Connection conn = null;
      Statement stmt = null;

      try {
         // Register JDBC driver

         try {
            Class.forName("com.mysql.jdbc.Driver");
         } catch (ClassNotFoundException e) {

         }

         // Open a connection
         System.out.println("Connecting to database...");
         conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mysql",
"root", "");

         // Execute a query
         System.out.println("Creating statement...");
         stmt = conn.createStatement();
         String sql;
         ResultSet rs;

         // Query the database for the desired data
         System.out.println("List of students who joined in 2018:");
         sql = "SELECT * FROM student1 WHERE registernumber LIKE '18%'";
```

```
            rs = stmt.executeQuery(sql);
            while (rs.next()) {
                System.out.println(rs.getString("name") + " (" +
rs.getString("registernumber") + ")");
            }
            System.out.println();

            System.out.println("List of students whose age is between 18-20:");
            sql = "SELECT * FROM student1 WHERE age BETWEEN 18 AND 20";
            rs = stmt.executeQuery(sql);
            while (rs.next()) {
                System.out.println(rs.getString("name") + " (" +
rs.getString("registernumber") + ")");
            }
            System.out.println();

            System.out.println("List of students with CGPA less than 5:");
            sql = "SELECT * FROM student1 WHERE cgpa < 5.0";
            rs = stmt.executeQuery(sql);
            while (rs.next()) {
                System.out.println(rs.getString("name") + " (" +
rs.getString("registernumber") + ")");
            }
            System.out.println();

            System.out.println("List of students who stay in hostel:");
            sql = "SELECT * FROM student1 WHERE hosteller = true";
            rs = stmt.executeQuery(sql);
            while (rs.next()) {
                System.out.println(rs.getString("name") + " (" +
rs.getString("registernumber") + ")");
            }
            System.out.println();
```

```
            System.out.println("List of 2019 batch students who are dayscholars:");
            sql = "SELECT * FROM student1 WHERE dayscholar = true AND
registernumber LIKE '19%'";
            rs = stmt.executeQuery(sql);
            while (rs.next()) {
               System.out.println(rs.getString("name") + " (" +
rs.getString("registernumber") + ")");
            }
            System.out.println();

            // Clean up resources
            rs.close();
            stmt.close();
            conn.close();
         } catch (SQLException se) {
// Handle errors for JDBC
            se.printStackTrace();
         } catch (Exception e) {
// Handle errors for Class.forName
            e.printStackTrace();
         } finally {
// Finally block used to close resources
            try {
               if (stmt != null) {
                  stmt.close();
               }
            } catch (SQLException se2) {
            } // nothing we can do
            try {
               if (conn != null) {
                  conn.close();
               }
            } catch (SQLException se) {
               se.printStackTrace();
```

```
            } // end finally try
        } // end try
    }
}
```

OUTPUT :