TRINITY INTERNATIONAL COLLEGE

(Tribhuvan University Affiliated)

Lab Assignment 7: Advance Java Programming

Submitted By:

Name: Dipendra Shrestha

Program: **B. Sc. (CSIT)**

Roll No: 16

Semester: seventh (7th)

Date: 5th July 2020

Submitted to:

Aman Maharjan

KATHMANDU, NEPAL
2020

1. Assume that a database named Astronomy contains the name and mass of each of the 8 planet of the solar system and its distance from the sun in a table with the schema planets (id, planet, mass, distance). Create a 2D array capable of holding all the data contained in this table. Use JDBC to populate the array from the data in the table. The 2D array should also hold the calculated value of gravitational force between each planet and the sun. This can be calculated using the data you just retrieved. Display the data in the final 2D array in a Swing JTable component with appropriate column headers.
   Use Newton's Law of Universal Gravitation to calculate the gravitational force between the sun and each of the planets (one at a time):

$$F = G\frac{m_s\, m_p}{r^2}$$

where,

gravitation constant (G) = 6.67430×10^{-11}$m^3kg^{-1}s^{-2}$,

the mass of the sun is 1.989×1030 kg

This can be represented in Java with the following code:

double G = 6.67430e-11;

double m_s = 1.989e+30;

⇨

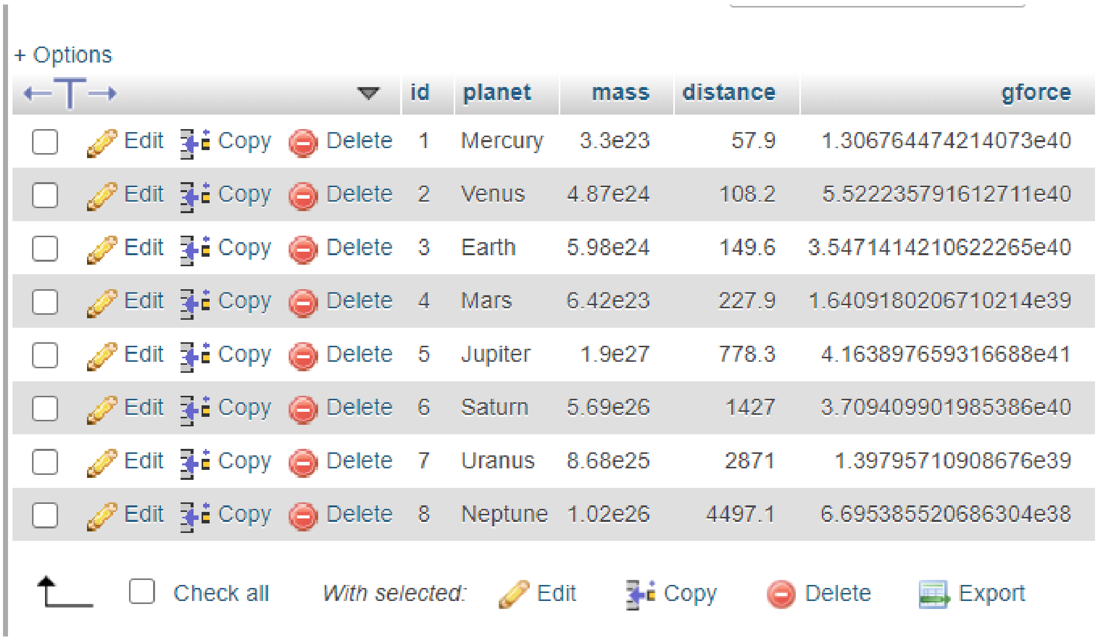
Program

```java
package combinedastronomy;

import javax.swing.*;
import java.sql.*;

public class CombinedAstronomy extends JFrame
{
    public static void main(String[] args) throws Exception
    {
        CombinedAstronomy frame = new CombinedAstronomy();
        frame.setVisible(true);

    }

    public CombinedAstronomy() throws Exception
    {
        String url = "jdbc:mariadb://localhost:3306/Astronomy";
        String username = "root";
        String password= "";
        Connection connection DriverManager.getConnection
                                             (url,username,password);

        String sql = "select mass , distance from planets";
        Statement statement = connection.createStatement();
        ResultSet resultSet = statement.executeQuery(sql);
```

```
//* Retrieving the stored value of mass and distance from Database *
        Double[] massOfPlanet = new Double[8];
        Double[] distance = new Double[8];
        int i = 0;
        while (resultSet.next())
        {
            massOfPlanet[i] = resultSet.getDouble("mass");
            distance[i] = resultSet.getDouble("distance");
            i++;
        }

        //******** Calculating Gravitational Force *****
        Double G = 6.67430e-11;
        Double ms = 1.989e+30;
        Double[] force = new Double[8];

        //calculating gravitational force and storing it into array.
        for (i = 0; i<force.length; i++)
        {
         force[i] = (G*ms*massOfPlanet[i])/(Math.pow(distance[i],2));
        }

        //*** Send Calculated Gravitational Force to database ****
        String insertSql = "UPDATE planets SET gforce = ? WHERE id =
                                                                (?)";
        PreparedStatement statement1 = connection.prepareStatement
                                                         (insertSql);
        for (i=1; i<=force.length; i++)
        {
            statement1.setDouble(1,force[i-1]);
            statement1.setInt(2,i);
            statement1.executeUpdate();
        }

//** Retrieving overall data of Database along with Calculated
      gravitational force in 2D Array.****

        String selectAllSql = "select * from planets";
        Statement statement2 = connection.createStatement();
        ResultSet resultSet1 = statement2.executeQuery(selectAllSql);

        Object[][] planetDetails = new Object[8][5];
        int row =0;
        while (resultSet1.next())
        {
            planetDetails[row][0] =resultSet1.getInt(1);
            planetDetails[row][1] = resultSet1.getString(2);
            planetDetails[row][2] = resultSet1.getDouble(3);
            planetDetails[row][3] = resultSet1.getDouble(4);
            planetDetails[row][4] = resultSet1.getDouble(5);

            row++;
        }
        statement.close();
        connection.close();

        //****** Display Values in Table ******************
        String column[] = {"ID", "Planets", "Mass", "Distance",
                           "Gravitational Force Between Plane and
```

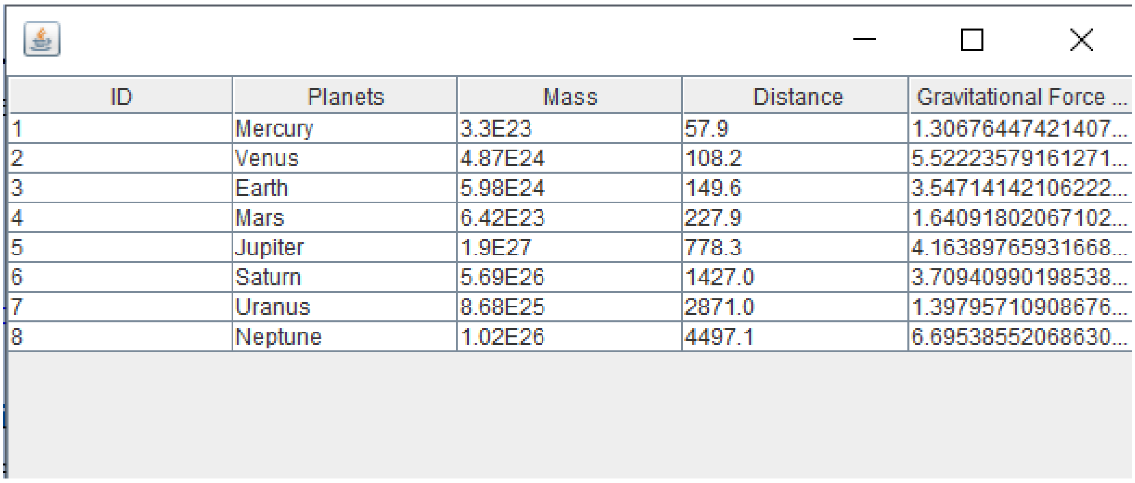
```
                                  Sun"};

            JTable table = new JTable(planetDetails, column)
            {
                public boolean isCellEditable(int row, int column)
                {
                    return false;
                }
            };
            JScrollPane pane = new JScrollPane(table);
            add(pane);

            pack();
            setDefaultCloseOperation(EXIT_ON_CLOSE);
        }
}
```

Output:



+ Options

←T→			id	planet	mass	distance	gforce
□	Edit	Copy	Delete	1	Mercury	3.3e23	57.9	1.306764474214073e40
□	Edit	Copy	Delete	2	Venus	4.87e24	108.2	5.522235791612711e40
□	Edit	Copy	Delete	3	Earth	5.98e24	149.6	3.547141421062265e40
□	Edit	Copy	Delete	4	Mars	6.42e23	227.9	1.640918020671021e39
□	Edit	Copy	Delete	5	Jupiter	1.9e27	778.3	4.163897659316688e41
□	Edit	Copy	Delete	6	Saturn	5.69e26	1427	3.709409901985386e40
□	Edit	Copy	Delete	7	Uranus	8.68e25	2871	1.39795710908676e39
□	Edit	Copy	Delete	8	Neptune	1.02e26	4497.1	6.695385520686304e38



ID	Planets	Mass	Distance	Gravitational Force ...
1	Mercury	3.3E23	57.9	1.30676447421407...
2	Venus	4.87E24	108.2	5.52223579161271...
3	Earth	5.98E24	149.6	3.54714142106222...
4	Mars	6.42E23	227.9	1.64091802067102...
5	Jupiter	1.9E27	778.3	4.16389765931668...
6	Saturn	5.69E26	1427.0	3.70940990198538...
7	Uranus	8.68E25	2871.0	1.39795710908676...
8	Neptune	1.02E26	4497.1	6.69538552068630...