Name : Harsh Raj , Group / Section : 619-A , UID : 22BCS11890 , Experiment – 05

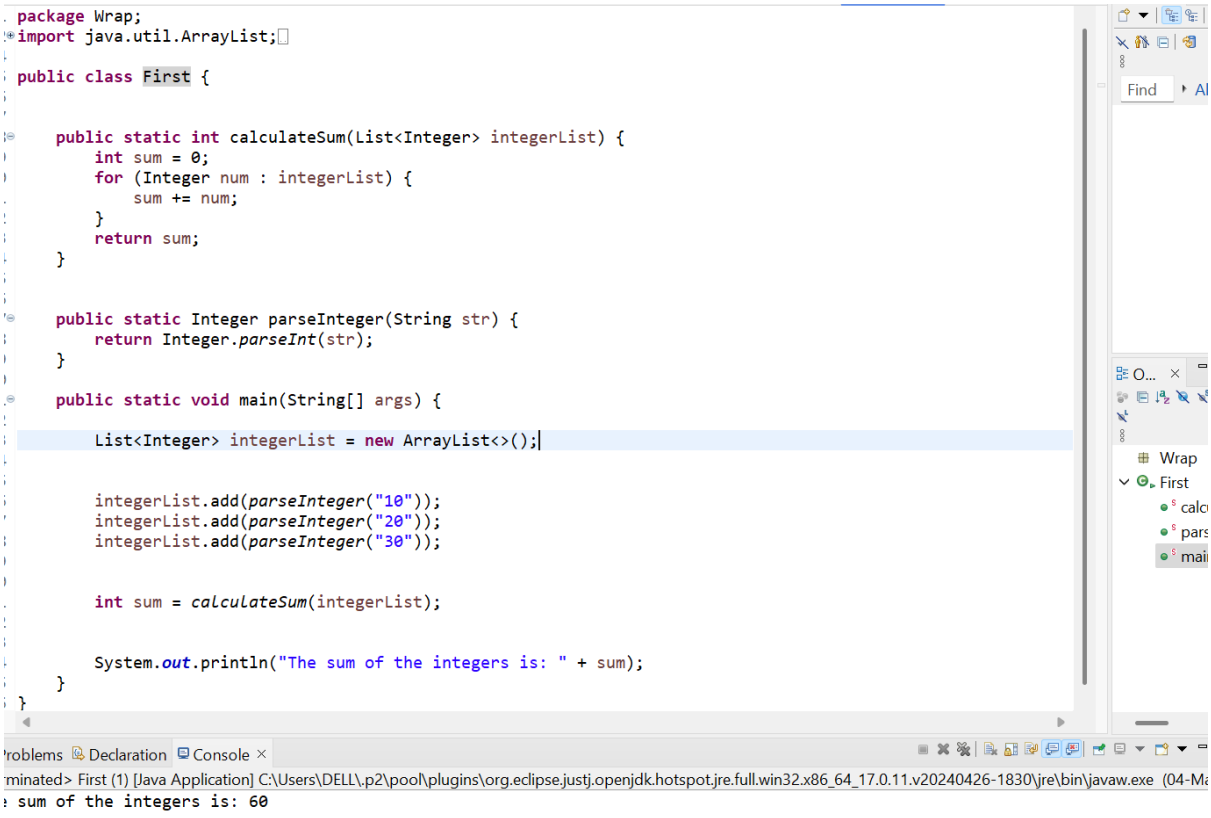
CODE

```
package Wrap;
import java.util.ArrayList;
import java.util.List;

public class First {


    public static int calculateSum(List<Integer> integerList) {
        int sum = 0;
        for (Integer num : integerList) {
            sum += num;
        }
        return sum;
    }


    public static Integer parseInteger(String str) {
        return Integer.parseInt(str);
    }

    public static void main(String[] args) {

        List<Integer> integerList = new ArrayList<>();


        integerList.add(parseInteger("10"));
        integerList.add(parseInteger("20"));
        integerList.add(parseInteger("30"));
```

Name : Harsh Raj , Group / Section : 619-A , UID : 22BCS11890 , Experiment – 05

```
        int sum = calculateSum(integerList);


        System.out.println("The sum of the integers is: " + sum);
    }
}
```

OUTPUT



CODE

```
package name;


import java.sql.*;
```

Name : Harsh Raj , Group / Section : 619-A , UID : 22BCS11890 , Experiment – 05

```
public class Mydb {
    private static final String URL = "jdbc:mysql://localhost:3306/amitdb";
    private static final String USER = "root";
    private static final String PASSWORD = "hraj72888@G";
    private Connection connection;



    public Mydb() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection = DriverManager.getConnection(URL, USER, PASSWORD);
            System.out.println("✅ Database Connected Successfully!");
        } catch (ClassNotFoundException e) {
            System.err.println("❌ JDBC Driver not found! Add MySQL Connector JAR.");
        } catch (SQLException e) {
            System.err.println("❌ Connection failed! Check credentials. Error: " + e.getMessage());
        }
    }


    public void getAllEmployees() {
        String query = "SELECT * FROM employee";
        try (Statement stmt = connection.createStatement();
             ResultSet rs = stmt.executeQuery(query)) {
            System.out.println("\n📌 Employee List:");
            while (rs.next()) {
                System.out.println("ID: " + rs.getInt("id") +
                                   ", Name: " + rs.getString("name") +
                                   ", Salary: " + rs.getDouble("salary"));
            }
        } catch (SQLException e) {
```

```
            System.err.println("❌ Error fetching employees: " + e.getMessage());
        }
    }


    public void addEmployee(int id,String name, double salary) {
        String query = "INSERT INTO employee (id,name, salary) VALUES (?,?, ?)";
        try (PreparedStatement stmt = connection.prepareStatement(query)) {
             stmt.setInt(1, id);
            stmt.setString(2, name);
            stmt.setDouble(3, salary);
            stmt.executeUpdate();
            System.out.println("✅ Employee added successfully: " + name);
        } catch (SQLException e) {
            System.err.println("❌ Error adding employee: " + e.getMessage());
        }
    }


    public void removeDuplicateEmployees() {
        String query = "DELETE e1 FROM employee e1 " +
                       "INNER JOIN employee e2 " +
                       "ON e1.name = e2.name AND e1.salary = e2.salary " +
                       "WHERE e1.id > e2.id";
        try (Statement stmt = connection.createStatement()) {
            int rowsDeleted = stmt.executeUpdate(query);
            System.out.println("🗑 " + rowsDeleted + " duplicate records removed.");
        } catch (SQLException e) {
            System.err.println("❌ Error removing duplicates: " + e.getMessage());
        }
```

```
    }


    public void closeConnection() {
        if (connection != null) {
            try {
                connection.close();
                System.out.println("✅ Database Connection Closed.");
            } catch (SQLException e) {
                System.err.println("❌ Error closing connection: " + e.getMessage());
            }
        }
    }


    public static void main(String[] args) {
        Mydb db = new Mydb();


        db.addEmployee(109,"James", 5000);
        db.addEmployee(110,"Harry Potter", 6000);
        db.addEmployee(111,"John cena", 5500);
        db.addEmployee(112,"Rahul", 8000);
        db.addEmployee(113,"Ram", 5200);
        db.addEmployee(140,"Devi", 6000);


        db.getAllEmployees();


        db.removeDuplicateEmployees();
```

```
        db.getAllEmployees();


        db.closeConnection();
    }
}
```

OUTPUT