Task – 3

```
import javax.persistence.*;


@Entity
@Table(name = "students")
public class Student {
   @Id
   @GeneratedValue(strategy = GenerationType.IDENTITY)
   private int id;


   private String name;
   private int age;
   private String grade;


   // Getters and setters
}
```

2.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;


public class CRUDOperationsExample {
    public static void main(String[] args) {
        String jdbcUrl = "jdbc:mysql://localhost:3306/university";
        String username = "your_username";
        String password = "your_password";


        try {
            Connection connection = DriverManager.getConnection(jdbcUrl, username,
password);

            // CREATE operation
            String createSQL = "INSERT INTO students (name, age, grade) VALUES (?, ?,
?)";
            PreparedStatement createStatement =
connection.prepareStatement(createSQL);
            createStatement.setString(1, "Carol");
            createStatement.setInt(2, 22);
            createStatement.setString(3, "Junior");
            createStatement.executeUpdate();
            System.out.println("Record created successfully");

            // READ operation
            String readSQL = "SELECT * FROM students";
            PreparedStatement readStatement = connection.prepareStatement(readSQL);
            ResultSet resultSet = readStatement.executeQuery();
            while (resultSet.next()) {
                System.out.println("ID: " + resultSet.getInt("id"));
                System.out.println("Name: " + resultSet.getString("name"));
                System.out.println("Age: " + resultSet.getInt("age"));
                System.out.println("Grade: " + resultSet.getString("grade"));
                System.out.println();
            }

            // UPDATE operation
            String updateSQL = "UPDATE students SET grade = ? WHERE name = ?";
            PreparedStatement updateStatement =
connection.prepareStatement(updateSQL);
            updateStatement.setString(1, "Senior");
            updateStatement.setString(2, "Alice");
            updateStatement.executeUpdate();
            System.out.println(“Record updated successfully”);

            // DELETE operation
            String deleteSQL = “DELETE FROM students WHERE name = ?”;
            PreparedStatement deleteStatement =
connection.prepareStatement(deleteSQL);
            deleteStatement.setString(1, “Bob”);
            deleteStatement.executeUpdate();
            System.out.println(“Record deleted successfully”);

            // Close resources
            createStatement.close();
            readStatement.close();
            updateStatement.close();
            deleteStatement.close();
            connection.close();

        }
    }
}
```