

Isaac Shoebottom, 3429069
Lab 8

Question 1:

```
import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.Scanner;

public class Question1 {

    public static void main(String[] args) {

        Question1 question1 = new Question1();

        Connection conn = question1.openConnection();

        Scanner sc = new Scanner(System.in);

        System.out.print("What is the name of the student?: ");

        String studentName = sc.nextLine();

        System.out.print("What is the email address of the student?: ");

        String emailAddress = sc.nextLine();

        System.out.print("What is the high school average of the student?: ");

        String highSchoolAverage = sc.nextLine();

        try {

            String query = "{CALL AcceptStudent(?, ?, ?)}";

            CallableStatement statement = conn.prepareCall(query);

            statement.setString(1, studentName);

            statement.setString(2, emailAddress);

            statement.setString(3, highSchoolAverage);

            statement.executeQuery();

        } catch (SQLException sqlException) {

            System.err.println(sqlException.getMessage());

        }

    }

}
```

```

        System.out.println(studentName + " with email address: " + emailAddress + "
and a high school average of: " + highSchoolAverage + " added");

        question1.closeConnection(conn);

    }

    private Connection openConnection() {
        final String url = "jdbc:mysql://cs1103.cs.unb.ca:3306/ishoebot";
        final String user = "ishoebot";
        final String password = "k7F0LH5B";
        Connection conn = null;
        try {
            conn = DriverManager.getConnection(url, user, password);
        }
        catch (Exception exception) {
            System.err.printf("Couldn't open a connection: (%s)",
exception.getMessage());
        }
        return conn;
    }

    private void closeConnection(Connection conn) {
        try {
            conn.close();
        }
        catch (Exception exception) {
            System.err.printf("Couldn't close connection: (%s)",
exception.getMessage());
        }
    }
}

```

Question 2:

Procedure:

```
CREATE PROCEDURE `studentTranscript`(  
    IN `ID` INT  
)  
BEGIN  
    IF EXISTS(SELECT 1 from Students where studentID = ID) THEN  
        SELECT courseName, letterGrade  
        FROM Enrollments  
        NATURAL JOIN Courses  
        NATURAL JOIN Students  
        WHERE studentID = ID;  
    ELSE  
        SIGNAL SQLSTATE '45006'  
        SET MESSAGE_TEXT = 'Student ID does not exist in system.';  
    END IF;  
END
```

Source Code:

```
import java.sql.*;

public class Question2 {

    public static void main(String[] args) {
        if(args.length < 1) {
            System.err.println("Usage: Question2 <StudentID>");
            System.exit(-1);
        }
        Question2 question2 = new Question2();
        Connection conn = question2.openConnection();
        String studentID = args[0];
        try {
            String query = "{CALL studentTranscript(?)}";
            CallableStatement statement = conn.prepareCall(query);
            statement.setString(1, studentID);
            ResultSet rs = statement.executeQuery();
            while(rs.next()) {
                String row = rs.getString(1) + ": " + rs.getString(2);
                System.out.println(row);
            }
        } catch (SQLException sqlException) {
            System.err.println(sqlException.getMessage());
        }
        question2.closeConnection(conn);
    }

    private Connection openConnection() {
        final String url = "jdbc:mysql://cs1103.cs.unb.ca:3306/ishoebot";
        final String user = "ishoebot";
        final String password = "k7F0LH5B";
```

```
        Connection conn = null;

        try {

            conn = DriverManager.getConnection(url, user, password);

        }

        catch (Exception exception) {

            System.err.printf("Couldn't open a connection: (%s)",
exception.getMessage());

        }

        return conn;

    }

    private void closeConnection(Connection conn) {

        try {

            conn.close();

        }

        catch (Exception exception) {

            System.err.printf("Couldn't close connection: (%s)",
exception.getMessage());

        }

    }

}
```