SCHOOL OF INFORMATION TECHNOLOGY

ASSIGNMENT COVER SHEET

WelTec
Wellington Institute of Technology
Te Whare Wānanga o te Awakairangi

PLEASE COMPLETE AND ATTACH TO YOUR ASSIGNMENT

First Name Francois **Last Name** Mindiel	**Student ID** **2804582**

Programme	**Bachelor of Information Technology**

Course	**Programming III**	**Group**

Assessment	**Assignment 2**	**Due Date 16/06/17**

Lecturer / Tutor	**Chalinor Baliuag**

Student Signature	

Student Declaration:* *This assignment is my own work.	***Received by the IT School on:* *(date stamp)***

✂--

Students please complete the bottom section as it is retained by the School of IT as proof that your assignment was handed in.

First Name:	**Francois**	
Last Name:	**Mindiel**	
Student ID:	**2804582**	
Programme and Course Code:	**BIT IT7374**	

Received by the IT School on: (date stamp)

Table of Contents

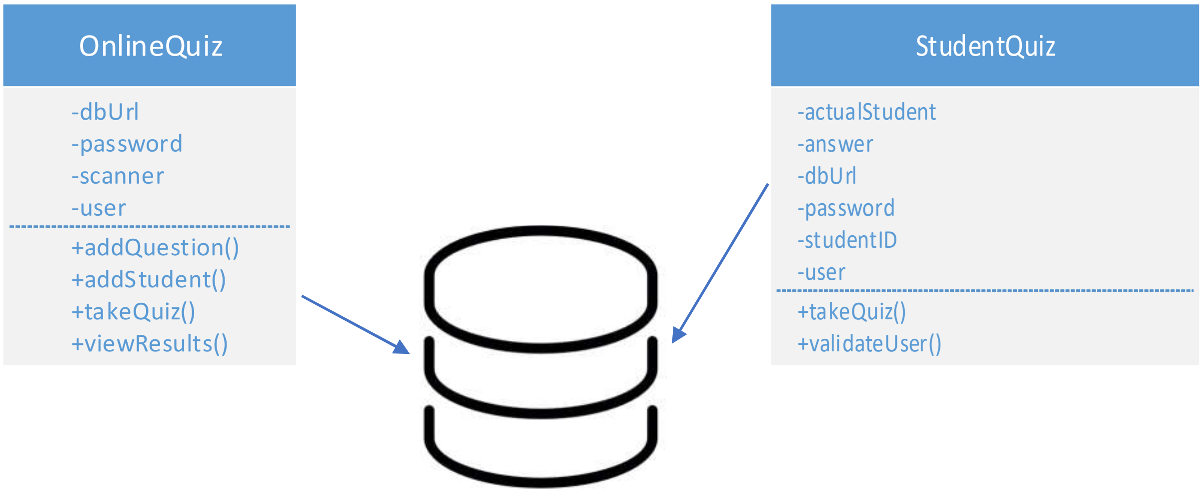
Introduction

This assignment is to demonstrate some of the skills learned during the course. For this demonstration, a simple online test for students will be designed, with access to a database to store students' profiles and their results, as well as the questions for the online test.

There will be another class for the teacher's access, which can be on a separate package or folder, as long as both the students' class and teacher's class point to the same database. The teacher's class will enable him to add students' information to the database, view their results, add questions, and try out the test as a check.

The students' class will only enable them to identify themselves, and take the test. Their results will then be stored in the database.

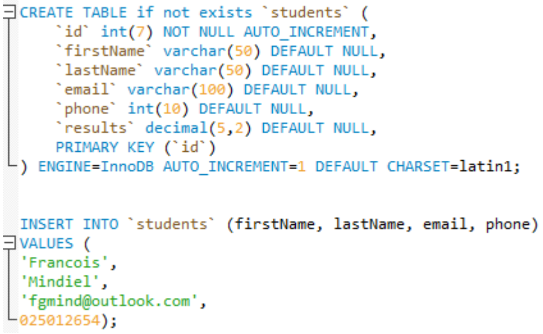
Class Diagram



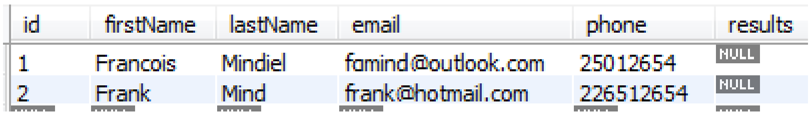
As can be seen from the diagram, both classes are separate from each other, but both can access the database.

Creating the database

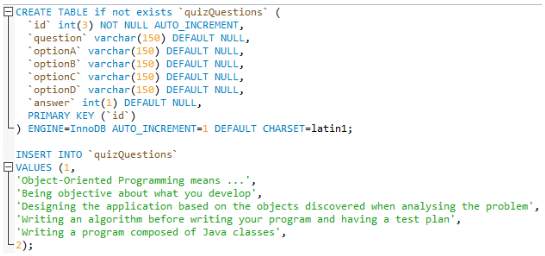
The first step in the development of the online test system is to establish the database. After installing MySQL WorkBench, and importing it into the runtime environment of the IDE, a script is run to populate the questions database, and students.



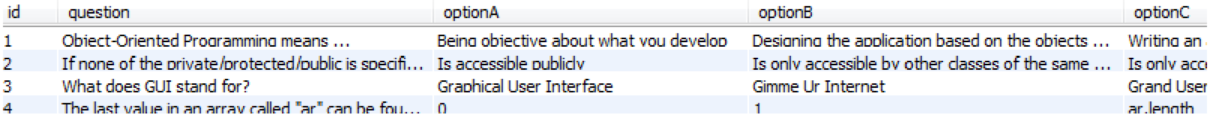
```
CREATE TABLE if not exists `students` (
    `id` int(7) NOT NULL AUTO_INCREMENT,
    `firstName` varchar(50) DEFAULT NULL,
    `lastName` varchar(50) DEFAULT NULL,
    `email` varchar(100) DEFAULT NULL,
    `phone` int(10) DEFAULT NULL,
    `results` decimal(5,2) DEFAULT NULL,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;


INSERT INTO `students` (firstName, lastName, email, phone)
VALUES (
'Francois',
'Mindiel',
'fgmind@outlook.com',
025012654);
```



id	firstName	lastName	email	phone	results
1	Francois	Mindiel	fgmind@outlook.com	25012654	NULL
2	Frank	Mind	frank@hotmail.com	226512654	NULL



```
CREATE TABLE if not exists `quizQuestions` (
  `id` int(3) NOT NULL AUTO_INCREMENT,
  `question` varchar(150) DEFAULT NULL,
  `optionA` varchar(150) DEFAULT NULL,
  `optionB` varchar(150) DEFAULT NULL,
  `optionC` varchar(150) DEFAULT NULL,
  `optionD` varchar(150) DEFAULT NULL,
  `answer` int(1) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;

INSERT INTO `quizQuestions`
VALUES (1,
'Object-Oriented Programming means ...',
'Being objective about what you develop',
'Designing the application based on the objects discovered when analysing the problem',
'Writing an algorithm before writing your program and having a test plan',
'Writing a program composed of Java classes',
2);
```



id	question	optionA	optionB	optionC
1	Object-Oriented Programming means ...	Being objective about what you develop	Designing the application based on the objects ...	Writing an
2	If none of the private/protected/public is specifi...	Is accessible publicly	Is only accessible by other classes of the same ...	Is only acc
3	What does GUI stand for?	Graphical User Interface	Gimme Ur Internet	Grand User
4	The last value in an array called "ar" can be fou...	0	1	ar.length

Then the OnlineQuizz class, accessible by the teacher is created, with the following capabilities:

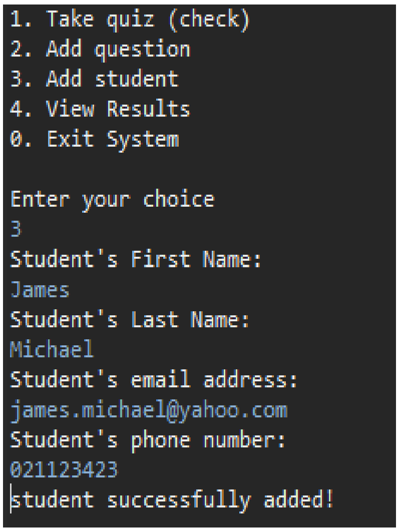
- Add students
- Add questions
- View results
- Take the quiz

The StudentQuizz class is given a different access name for the database, and can only take the test after validation of their student ID.

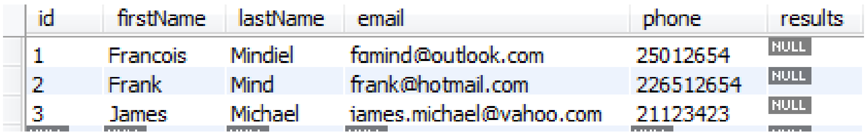
Their record is updated as soon as they have completed the test.

OnlineQuizz Class

Adding a student



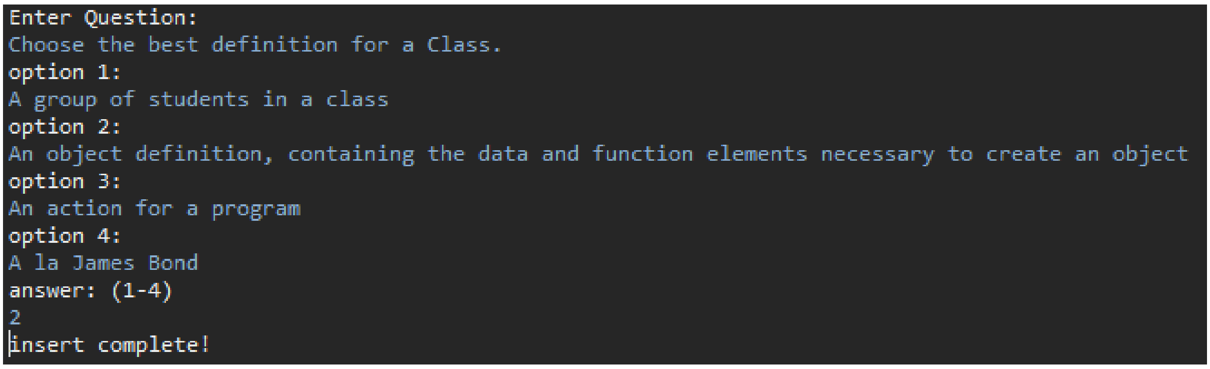
```
1. Take quiz (check)
2. Add question
3. Add student
4. View Results
0. Exit System

Enter your choice
3
Student's First Name:
James
Student's Last Name:
Michael
Student's email address:
james.michael@yahoo.com
Student's phone number:
021123423
student successfully added!
```



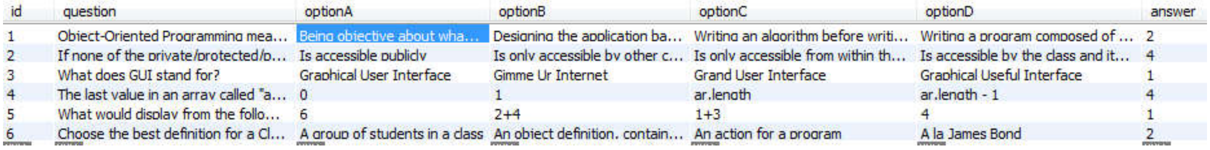
id	firstName	lastName	email	phone	results
1	Francois	Mindiel	famind@outlook.com	25012654	NULL
2	Frank	Mind	frank@hotmail.com	226512654	NULL
3	James	Michael	james.michael@yahoo.com	21123423	NULL

Since the student has just been added, there are no results displayed for him yet.

Adding a question

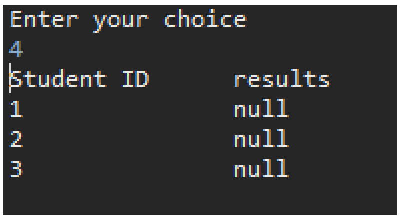


```
Enter Question:
Choose the best definition for a Class.
option 1:
A group of students in a class
option 2:
An object definition, containing the data and function elements necessary to create an object
option 3:
An action for a program
option 4:
A la James Bond
answer: (1-4)
2
insert complete!
```



id	question	optionA	optionB	optionC	optionD	answer
1	Object-Oriented Programming mea...	Being objective about wha...	Designing the application ba...	Writing an algorithm before writi...	Writing a program composed of ...	2
2	If none of the private/protected/p...	Is accessible publicly	Is only accessible by other c...	Is only accessible from within th...	Is accessible by the class and it...	4
3	What does GUI stand for?	Graphical User Interface	Gimme Ur Internet	Grand User Interface	Graphical Useful Interface	1
4	The last value in an array called "a...	0	1	ar.length	ar.length - 1	4
5	What would display from the follo...	6	2+4	1+3	4	1
6	Choose the best definition for a Cl...	A group of students in a class	An object definition, contain...	An action for a program	A la James Bond	2

Viewing results

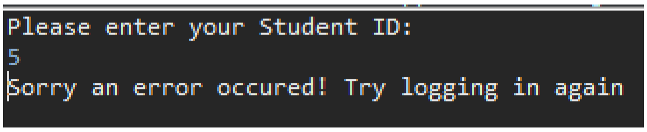


```
Enter your choice
4
Student ID        results
1                 null
2                 null
3                 null
```

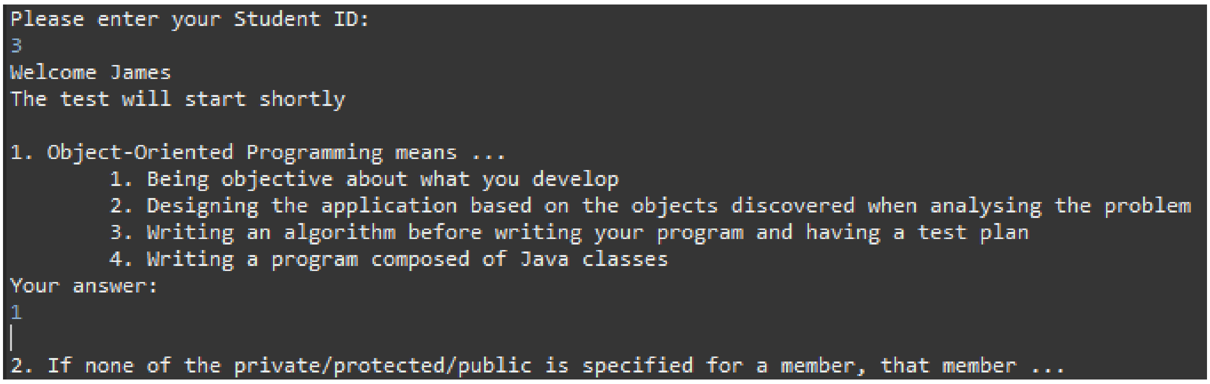
Since no student has taken the test yet, results are all 0.

StudentQuiz class

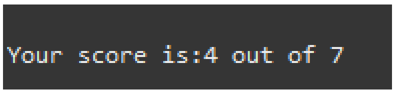
For this class, no option is given to the user. The student just has to enter his student id to access the test and begin it. If it's the wrong id, he cannot access it. Upon completion of test, he can see his score. In addition the system updates the database, and the teacher can now see the results from her class.



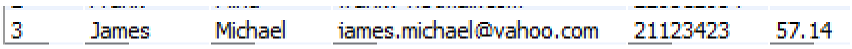
```
Please enter your Student ID:
5
Sorry an error occured! Try logging in again
```



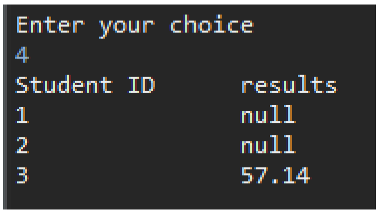
```
Please enter your Student ID:
3
Welcome James
The test will start shortly

1. Object-Oriented Programming means ...
        1. Being objective about what you develop
        2. Designing the application based on the objects discovered when analysing the problem
        3. Writing an algorithm before writing your program and having a test plan
        4. Writing a program composed of Java classes
Your answer:
1

2. If none of the private/protected/public is specified for a member, that member ...
```



```
Your score is:4 out of 7
```



3	James	Michael	iames.michael@vahoo.com	21123423	57.14



```
Enter your choice
4
Student ID      results
1               null
2               null
3               57.14
```

Conclusion

This is quite a simple approach to a distributed system in Java. The system could be further developed to have the test displayed on a webpage instead, for example. It does however accomplish the outcome, though this would have to have the database secured with a password for the users that want to connect to it; to avoid unauthorised access or cheating from the students.

Appendix

QuizQuestions.sql

```
create database if not exists onlineQuiz;

use onlineQuiz;

drop table if exists quizQuestions;

drop table if exists students;


CREATE TABLE if not exists `students` (

        `id` int(7) NOT NULL AUTO_INCREMENT,

        `firstName` varchar(50) DEFAULT NULL,

        `lastName` varchar(50) DEFAULT NULL,

        `email` varchar(100) DEFAULT NULL,

        `phone` int(10) DEFAULT NULL,

        `results` decimal(5,2) DEFAULT NULL,

        PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;


INSERT INTO `students` (firstName, lastName, email, phone)

VALUES (

'Francois',

'Mindiel',

'fgmind@outlook.com',

025012654);


INSERT INTO `students` (firstName, lastName, email, phone)

VALUES (

'Frank',

'Mind',

'frank@hotmail.com',

0226512654);


CREATE TABLE if not exists `quizQuestions` (

  `id` int(3) NOT NULL AUTO_INCREMENT,

  `question` varchar(150) DEFAULT NULL,

  `optionA` varchar(150) DEFAULT NULL,

  `optionB` varchar(150) DEFAULT NULL,

  `optionC` varchar(150) DEFAULT NULL,
  `optionD` varchar(150) DEFAULT NULL,

  `answer` int(1) DEFAULT NULL,

  PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;


INSERT INTO `quizQuestions`

VALUES (1,

'Object-Oriented Programming means ...',

'Being objective about what you develop',

'Designing the application based on the objects discovered when analysing the problem',

'Writing an algorithm before writing your program and having a test plan',

'Writing a program composed of Java classes',

2);


INSERT INTO `quizQuestions`

VALUES (2,

'If none of the private/protected/public is specified for a member, that member ...',

'Is accessible publicly',

'Is only accessible by other classes of the same package',

'Is only accessible from within the class',

'Is accessible by the class and its subclasses',

4);


INSERT INTO `quizQuestions`

VALUES (3,

'What does GUI stand for?',

'Graphical User Interface',

'Gimme Ur Internet',

'Grand User Interface',

'Graphical Useful Interface',

1);


INSERT INTO `quizQuestions`

VALUES (4,

'The last value in an array called "ar" can be found at index:',

'0',

'1',

'ar.length',
'ar.length - 1',
4);

INSERT INTO `quizQuestions`
VALUES (5,
'What would display from the following statements? int [ ] nums = {1,2,3,4,5,6}; System.out.println((nums[1] + nums[3]));',
'6',
'2+4',
'1+3',
'4',
1);
```

OnlineQuizz.java

```
package teacher;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Scanner;

public class OnlineQuiz {
    private String dbUrl = "jdbc:mysql://localhost:3306/onlineQuiz?useSSL=false";
    private String user = "root";
    private String password = "root";
    private Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        OnlineQuiz oq = new OnlineQuiz();
        boolean exit = false;

        while (exit == false){
            System.out.println("\n");
            System.out.println("1. Take quiz (check)");
            System.out.println("2. Add question");
            System.out.println("3. Add student");
            System.out.println("4. View Results");
            System.out.println("0. Exit System");

            System.out.println("\nEnter your choice");

            @SuppressWarnings("resource")

            Scanner sc = new Scanner(System.in);

            int choice = sc.nextInt();


            switch(choice){

            case 1:

                oq.takeQuiz();

                break;

            case 2:

                oq.addQuestion();

                break;

            case 3:

                oq.addStudent();

                break;

            case 4:

                oq.viewResults();

                break;

            case 0:

                exit = true;

                break;

            default:

                System.out.println("Error! Choice should be a number between 1 and 4! (or 0 to exit)");

                exit = false;

            }

        }

        System.out.println("\nGoodbye!");

    }


    private void takeQuiz(){

        int score = 0;

        int quesCount = 0;

        try {

//          1. Establishing Connection

            Connection myConn = DriverManager.getConnection(dbUrl, user, password);

//          2. Create a statement

            Statement myStmt = myConn.createStatement();
```

```
//          3. Execute the query
            ResultSet myRs = myStmt.executeQuery("select * from quizQuestions");
//          4. Process the resultSet object
            while(myRs.next()){
                System.out.println("\n" + myRs.getString("id") + ". " + myRs.getString("question"));
                System.out.println("\t1. " + myRs.getString("optionA"));
                System.out.println("\t2. " + myRs.getString("optionB"));
                System.out.println("\t3. " + myRs.getString("optionC"));
                System.out.println("\t4. " + myRs.getString("optionD"));


                System.out.println("Your answer: ");
                int tempScore = scanner.nextInt();


                if (tempScore == myRs.getInt("answer")){
                    score++;
                }
                quesCount++;
            }
            System.out.println("\n\n\n\n" + "Your score is:" + score + " out of " + quesCount);


//          5. Close statement & connection
            myStmt.close();
            myConn.close();
        }
        catch (Exception e) {
            e.getStackTrace();
        }
    }


    private void addQuestion(){
        try {
//          1. Establishing Connection
            Connection myConn = DriverManager.getConnection(dbUrl, user, password);
//          2. Create a statement
            Statement myStmt = myConn.createStatement();


//          Creating Question and answers
            System.out.println("Enter Question:");
            String question = scanner.nextLine();

            System.out.println("option 1: ");
            String optionA = scanner.nextLine();

            System.out.println("option 2: ");
            String optionB = scanner.nextLine();

            System.out.println("option 3: ");
            String optionC = scanner.nextLine();

            System.out.println("option 4: ");
            String optionD = scanner.nextLine();

            System.out.println("answer: (1-4) ");
            int answer = scanner.nextInt();

//          3. create mySql command / query
            String sql = "insert into quizQuestions"
                        + "(question, optionA, optionB, optionC, optionD, answer)"
                        + "values"
                        + "('" + question + "',"
                        + "'" + optionA + "',"
                        + "'" + optionB + "',"
                        + "'" + optionC + "',"
                        + "'" + optionD + "',"
                        + answer +");";

//          4. send command to mySql
            myStmt.executeUpdate(sql);

            System.out.println("insert complete!");

//          5. Close statement & connection
            myStmt.close();
            myConn.close();
        }
        catch (Exception e) {
```

```
                                        e.getStackTrace();
                        }
            }

            private void viewResults(){
                        try {
//                                      1. Establishing Connection
                                        Connection myConn = DriverManager.getConnection(dbUrl, user, password);
//                                      2. Create a statement
                                        Statement myStmt = myConn.createStatement();
//                                      3. Execute the query
                                        ResultSet myRs = myStmt.executeQuery("select * from students");
//                                      4. Process the resultSet object

                                        System.out.println("Student ID\tresults");
                                        while(myRs.next()){
                                                    System.out.print(myRs.getString("id"));
                                                    System.out.print("\t\t");
                                                    System.out.print(myRs.getString("results"));
                                                    System.out.print("\n");
                                        }

//                                      5. Close statement & connection
                                        myStmt.close();
                                        myConn.close();

                        }
                        catch (Exception e) {
                                        e.getStackTrace();
                        }
            }

            private void addStudent(){
                        try {
//                                      1. Establishing Connection
                                        Connection myConn = DriverManager.getConnection(dbUrl, user, password);
//                                      2. Create a statement
                                        Statement myStmt = myConn.createStatement();
                                        System.out.println("Student's First Name: ");
                                        String firstName = scanner.next();
                                        System.out.println("Student's Last Name: ");
                                        String lastName = scanner.next();
                                        System.out.println("Student's email address: ");
                                        String email = scanner.next();
                                        System.out.println("Student's phone number: ");
                                        int phone = scanner.nextInt();

//                      3. create mySql command / query
                                        String sql = "insert into students"
                                                                + "(firstName, lastName, email, phone)"
                                                                + "values"
                                                                + "('" + firstName + "',"
                                                                + "'" + lastName + "',"
                                                                + "'" + email + "',"
                                                                + phone + ");";

//                      4. send command to mySql
                                        myStmt.executeUpdate(sql);

                                        System.out.println("student successfully added!");

//                      5. Close statement & connection
                                        myStmt.close();
                                        myConn.close();
                        }
                        catch (Exception e) {
                                        e.getStackTrace();
                        }
            }
}
```

StudentQuizz.java

```
package students;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Scanner;

public class StudentQuiz {
        private String dbUrl = "jdbc:mysql://localhost:3306/onlineQuiz?useSSL=false";
        private String user = "student";
        private String password = "";
        private boolean actualStudent = false;
        private Scanner answer = new Scanner(System.in);
        private int studentID = 0;

        public static void main(String[] args) {

                StudentQuiz studentQuiz = new StudentQuiz();

                studentQuiz.validateUser();
                studentQuiz.takeQuiz();
        }

        private void validateUser(){
                System.out.println("Please enter your Student ID: ");
                studentID = answer.nextInt();

                try {
//                              1. Establishing Connection
                                Connection myConn = DriverManager.getConnection(dbUrl, user, password);
//                              2. Create a statement
                                Statement myStmt = myConn.createStatement();
//                              3. Execute the query
                                ResultSet myRs = myStmt.executeQuery("select * from students");
//                              4. Process the resultSet object
                                while (myRs.next()){
                                        if (myRs.getInt("id") == studentID){
```

```
                    System.out.println("Welcome " + myRs.getString("firstName"));
                    System.out.println("The test will start shortly");
                    actualStudent = true;
                }
            }
            if (actualStudent == false){
                System.out.println("Sorry an error occured! Try logging in again");
                System.exit(-1);
            }

//          5. Close statement & connection
            myStmt.close();
            myConn.close();
        }
        catch (Exception e) {
            e.getStackTrace();
        }
    }

    void takeQuiz(){

        int score = 0;
        int quesCount = 0;

        try {
//          1. Establishing Connection
            Connection myConn = DriverManager.getConnection(dbUrl, user, password);
//          2. Create a statement
            Statement myStmt = myConn.createStatement();
//          3. Execute the query
            ResultSet myRs = myStmt.executeQuery("select * from quizQuestions");
//          4. Process the resultSet object
            while(myRs.next()){
                System.out.println("\n" + myRs.getString("id") + ". " + myRs.getString("question"));
                System.out.println("\t1. " + myRs.getString("optionA"));
                System.out.println("\t2. " + myRs.getString("optionB"));
                System.out.println("\t3. " + myRs.getString("optionC"));
                System.out.println("\t4. " + myRs.getString("optionD"));
```

```
                                        System.out.println("Your answer: ");
                                        int tempScore = answer.nextInt();


                                        if (tempScore == myRs.getInt("answer")){
                                                score++;
                                        }
                                        quesCount++;
                                }
                                System.out.println("\n\n\n\n" + "Your score is:" + score + " out of " + quesCount);


                                String sqlAddResult = "UPDATE students SET results = " + (double)score/quesCount*100 + " WHERE id =" + studentID + ";";
                                myStmt.executeUpdate(sqlAddResult);


//                              5. Close statement & connection
                                myStmt.close();
                                myConn.close();


                        }
                        catch (Exception e) {
                                e.getStackTrace();
                        }
                }
}
```