The School of Technology

BSc (Hons) Computer Science (Software Engineering)

Module: 5CI022

Module Assesment

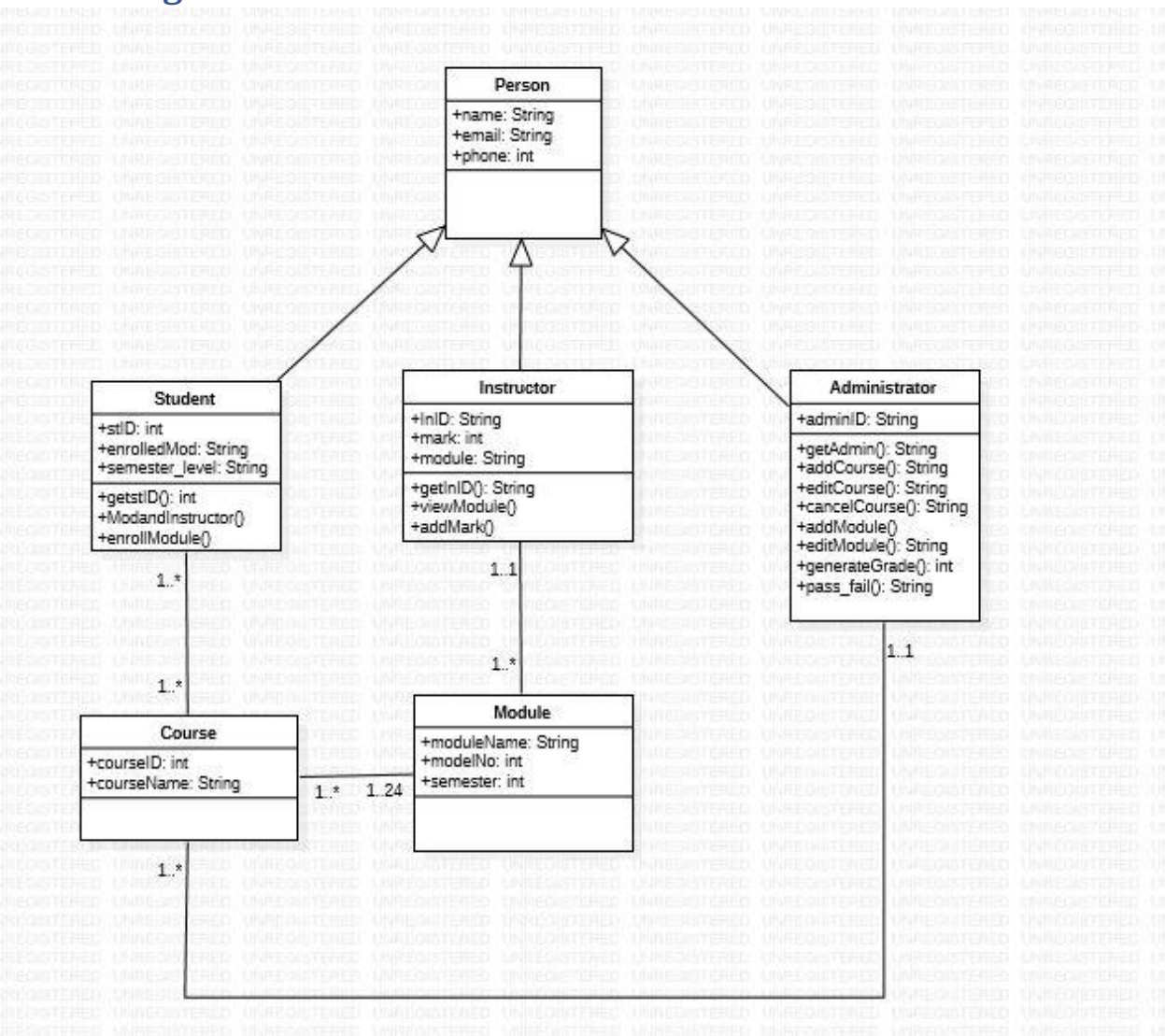
Module Leader: Ms. Sachini Vindya Gunasekara

Rashmika Jayasinghe (UOW 1900946)

Submission Date: 18.02.2020

Contents

UML Diagram



OOP Concepts and File Reading and File and Writing

The developed program for the assessment is using some object-oriented concepts such as Inheritance, Encapsulation , data hiding. Data hiding and encapsulation are used in the program when the system some variable are declared as public and some are declared as private , protected. Inheritance is used when the Administrator , Instructor and Student classes are sub classes of the Person super class.

Data Structures

In my program main data structure I used hashmap and it is used for data store. I also use two dimensional arrays it is also used for data store.

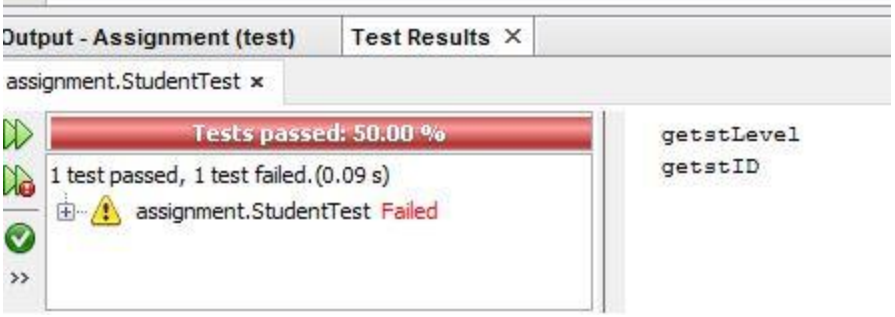
Junit Testing

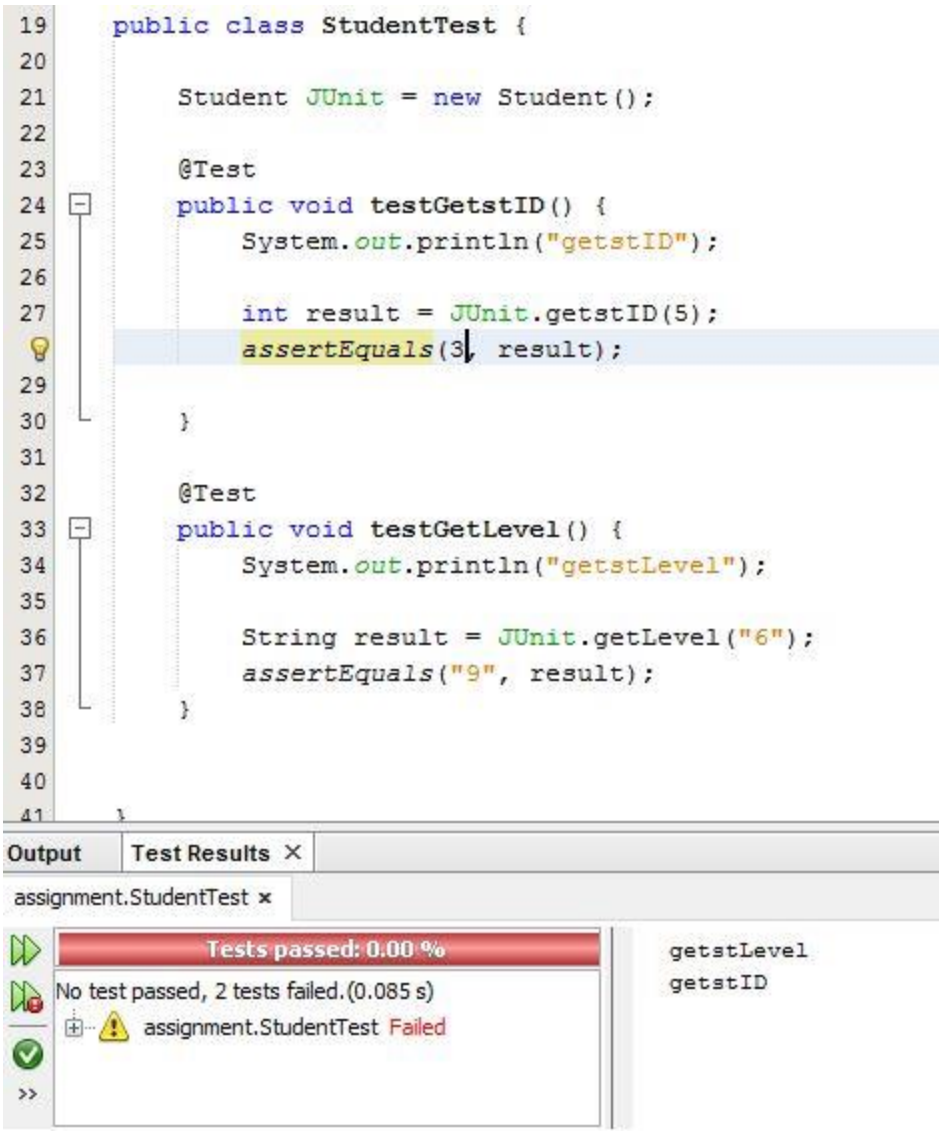
public void getstID() –get student ID as a input.

public void getLevel() - get semester level as a input.

```
19  public class StudentTest {
20
21      Student JUnit = new Student();
22
23      @Test
24      public void testGetstID() {
25          System.out.println("getstID");
26
27          int result = JUnit.getstID(5);
28          assertEquals(5, result);
29          // TODO review the generated test code and remove the default call to fail.
30          //fail("The test case is a prototype.");
31      }
32
```

```
23      @Test
24      public void testGetstID() {
25          System.out.println("getstID");
26
27          int result = JUnit.getstID(5);
28          assertEquals(5, result);
29
30      }
31
32      @Test
33      public void testGetLevel() {
34          System.out.println("getstLevel");
35
36          String result = JUnit.getLevel("6");
37          assertEquals("9", result);
38      }
39
40
41  }
42
```

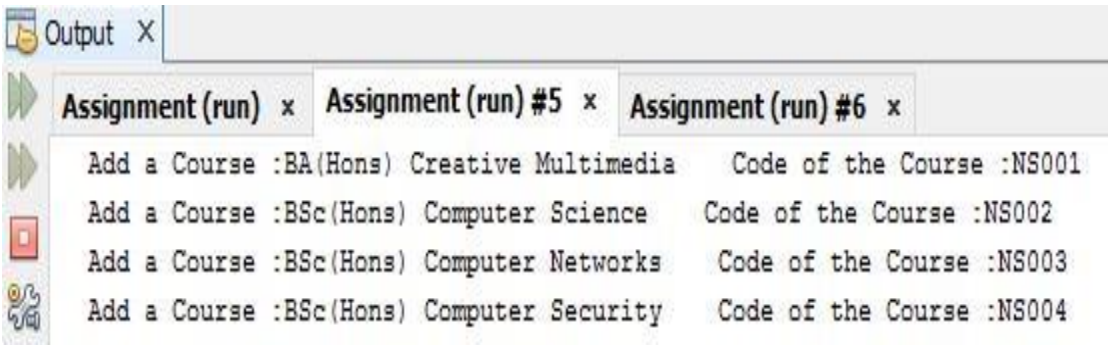




Classes

Administrator

- public void addCourse(String code, String courseName) – Add course code and course name



- protected void editCourse() - Edit course which already exists.

```
Code of Course to be Edited: NS002
New Course Name: BSc(Hons) Business Information System
```

- public void cancelCourse() – Cancel a course which already exists.

```
Cancelled Course: NS003
Code :NS003 Course :BSc(Hons) Computer Networks cancelled
```

- public void addModule(int moduleNo,String moduleName) –Add modules to the course.

```
Add modules to the Course :Communication Skills
Add modules to the Course :Information System
Add modules to the Course :Mathematics
Add modules to the Course :System Analysis Design
Add modules to the Course :Personal Computing
Add modules to the Course :Information System
```

- public void editModule(int moduleNo,String moduleName) -Edit modules which already exists.

```
Module edited :Database
Module edited :Programming
```

- public int generateGrade() - Enter average marks as a user input.

```
Enter student average marks of all modules :
67
Student Grade :B
Enter  Grade :
B
```

- public void pass_fail(String stdid) – Enter marks and grade as a user input and give which class that the student have.

```
Enter student average marks of all modules :
67
Student Grade :B
Enter  Grade :
B
First upper class !
```

Student

- public String getStID() – Enter student ID as a user input.

```
Enter Student ID :
ST001
```

public void enrollModule(String uId, String modCode,String module) –
Allows a student object to be created and to enroll for all modules

-

```
User ST001 successfully enrolled to Web Application
User ST001 successfully enrolled to Personal Computing
User ST001 successfully enrolled to Programming
User ST001 successfully enrolled to Database
User ST001 successfully enrolled to Mathematics
User ST001 successfully enrolled to Information System
```

- public void ModandInstructor(String enrolledModCode,String Instructor) -Allow a student which instructor that they have.

```
Module: NS001 Instructor: L1
Module: NS015 Instructor: L1
Module: NS018 Instructor: L2
Module: NS014 Instructor: L2
Module: NS013 Instructor: L3
```

Instructor

- public String getInID() – Get Instructor ID as user input.

```
Input Instructor ID :
L1
InStructor  ID :L1
Ruvii Jayasekara
ruvii@gmail.com
34562

Input Instructor ID :
L2
InStructor  ID :L2
Chamod Lokuliyanage
chamod@yahoo.com
98764

Input Instructor ID :
L3
InStructor  ID :L3
Michle Clarke
mike@gmail.com
98098
```

- public void viewModule(String InID, String modName) – Allow instructor to view modules that they can teach.

```
Modules Can view by Instructor L1 - Mathematics
Modules Can view by Instructor L1 - Information System
Modules Can view by Instructor L1 - Personal Computing
Modules Can view by Instructor L1 - Programming
Modules Can view by Instructor L1 - Web Application
Modules Can view by Instructor L1 - System Analysis Design


Modules Can view by Instructor L2 - System Analysis Design
Modules Can view by Instructor L2 - Information System
Modules Can view by Instructor L2 - Mathematics
Modules Can view by Instructor L2 - Web Application
Modules Can view by Instructor L2 - Personal Computing
Modules Can view by Instructor L2 - Programming


Modules Can view by Instructor L3 - Web Application
Modules Can view by Instructor L3 - Personal Computing
Modules Can view by Instructor L3 - Programming
Modules Can view by Instructor L3 - Information System
Modules Can view by Instructor L3 - Mathematics
```

- public void addMark() – Allow instructor to add marks for the modules

```
Pass mark of the Mathematics :40

Pass mark of the Information System :40

Pass mark of the Personal Computing :40

Pass mark of the Programming :40

Pass mark of the Web Application :40

Pass mark of the System Analysis Design :40
```

Course

- public Course(String courseID,String courseName) – Creates a course object.

```
Course Available
Course ID :NS001    Course Name :BA(Hons) Creative Multimedia
Course ID :NS002    Course Name :BSc(Hons) Computer Science
Course ID :NS003    Course Name :BSc(Hons) Computer Networks
Course ID :NS004    Course Name :BSc(Hons) Business Science
```

Module

- public Module(int modelNo,String moduleName,String semester) – Create a module object.

```
Module number :442 -- SEMESTER : One -- Module Name :Programming
Module number :441 -- SEMESTER : One -- Module Name :Database
Module number :444 -- SEMESTER : One -- Module Name :Personal Computing
Module number :551 -- SEMESTER : Two -- Module Name :Mathematics
Module number :552 -- SEMESTER : Two -- Module Name :Information System
Module number :553 -- SEMESTER : Two -- Module Name :System Analysis Design
```