APPENDIX B:

Declaration of Original Work for CE/CZ2002 Assignment

We hereby declare that the attached group assignment has been researched, undertaken, completed and submitted as a collective effort by the group members listed below.

We have honored the principles of academic integrity and have upheld Student Code of Academic Conduct in the completion of this work.

We understand that if plagiarism is found in the assignment, then lower marks or no marks will be awarded for the assessed work.

Name	Course (CE2002 or CZ2002)	Lab Group	Signature /Date

Important notes:

1. Name must **EXACTLY MATCH** the one printed on your Matriculation Card.

DESIGN CONSIDERATION

Students, professors and courses are specific and essential classes for the course and marks registration system. Since students and professors are people, they extend or inherit from the class Person which contains all personal details of the student or professor. In this case, person alone is meaningless to a course registration system, thus it is abstract. However, due to the need to distinguish each student and professor, the Person class defines an abstract method, getID() and setID(int i) to be implemented by the subclasses.

In differentiating students and professors, they need to have a unique ID. Therefore, they implements an interface called PrimaryKeyManager which requires the implementation of method autoIncrement(int i). This applies to Course and Group classes as well where ID is used to distinguish them. Besides the method, these classes need to define a static integer value, named pk, which starts from 1. The object ID of the above classes will be determined by this "pk" value, and this value will be incremented upon construction of the object by the above method. This ensures that the objects created are distinct with different IDs. In addition, to check whether or not the objects are equal, the 'equals' method is overridden by comparing the classes ('getClass') and ID values.

Student, Professor and Course are entity classes, and they are aggregated under School class. This is to ensure that when Student, Professor and Course objects are saved by Java Serializable, the objects will not be duplicated when loaded and the associations are still intact with the right reference. This ensures that when you change a course by using the professor's 'getCourse' method, the same course aggregated under School will also receive the change.

In determining course types, a CourseType enumeration class is used since the types will hardly change and thus act as a constant. These constants are used to determine whether or not a course contains lecture, tutorials or labs. A CourseGroup object acts as a control to manage the groups. Depending on the CourseType, CourseGroup will determine instantiate the right Group object for the course. Hence, Group objects are entities containing the students and capacity information.

In determining course assessment components, a key-value data structure, HashMap, is used to store the name of the assessment as the key and the weights as the value.

Calculation formula for total marks: $$Total\ marks = \frac{\sum marks_i weight_i}{total\ weights}$$

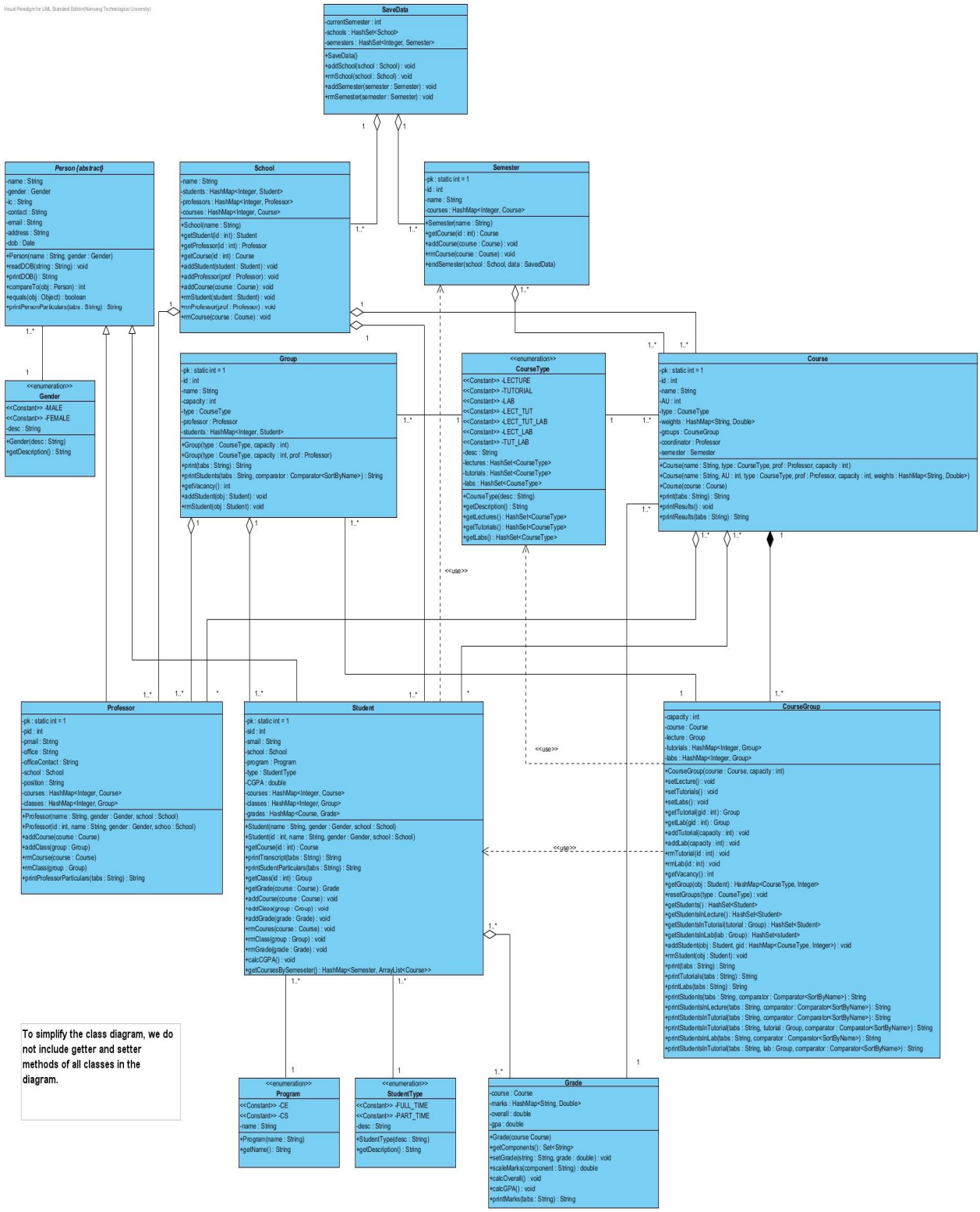
Registration of a student to a course occurs only in a new academic semester. Therefore, there is a need to create a new academic semester before registration. The Semester object will contain all courses that were added in by the administrator and are open for registration. Students are registered in by inputting their appropriate ID and tutorial and lab group IDs. In this application, every course can only have a maximum of 1 lecture group. During the academic semester, students' grades or marks for the final exam as well as other assessment components can also be input by specifying the student ID and course ID. Finally by ending the semester, the overall marks will be calculated and students' CGPA will be updated.

Therefore, there are 4 major parts of our application which involves students, professors, courses and semesters. Each of them will have a static main method defined to interact with the user, thus acting as a boundary method. However, these main methods cannot be invoked by running their class files, they can only be invoked by the application class SCRAME depending on the user inputs. Although the main methods are classified as boundaries, there exists some control logic to interact with each object's interface.

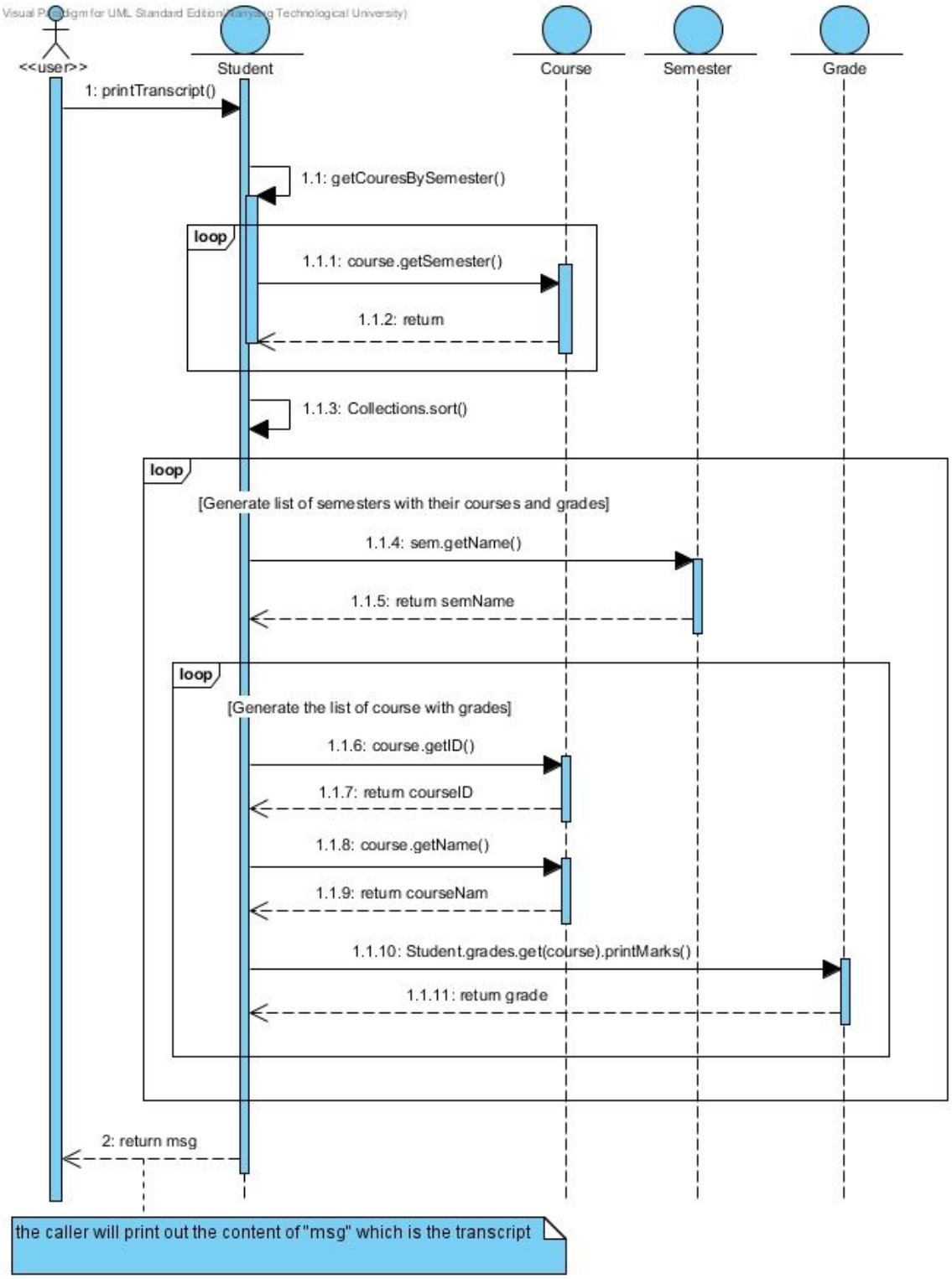
The use of polymorphism may be vague in our application classes due to the need of using and returning precise and accurate types in the program. For example, we cannot pass a Person casted object in registration of courses because professors are not students. However, some of the data structures, like HashMap, can be referred using the superclass Map, but was not shown in implementation for the sake of simplicity. One particular example of polymorphism is shown in Person's 'equals' method, which check for the class and cast it to Person type regardless of whether or not the object is of Student or Professor class.

```
@Override
public boolean equals(Object obj) {
	if (!(obj instanceof Person)) return false;
	Person person = (Person) obj;
	if (person.getClass().equals(this.getClass())) {
		if (person.getID() == this.getID()) return true;
		if (person.getName().equals(this.getName()) &&
			person.getGender() == this.getGender() ) return true;
	}
	return false;
	}
```

CLASS DIAGRAM



SEQUENCE DIAGRAM



TEST CASE

Test case 1: Student

- Create a new student
- Add an existing student

```
Choice: 4
Please input student name: Aloissus Chia
Please choose gender (M/F): m
Student 18, Aloissus Chia created successfully.

Students List:
NO    | STUDENT ID | NAME
--------------------------------------------------------------------------------
1     | 1          | Kenny Pang
2     | 2          | Lucas Neo
3     | 3          | Tan Yun Hwa
4     | 4          | Loh Jun Kiat
5     | 5          | Duong Canh Vi
6     | 6          | Melissa Teo
7     | 7          | Liu Wan Ling
8     | 8          | Victor Tan
9     | 9          | Duphk Dipth
10    | 10         | Linda Yeo
11    | 11         | Muhammad
12    | 12         | Wendy Koh
13    | 13         | Louis Tan
14    | 14         | Lois Chua
15    | 15         | Sean Ng
16    | 18         | Aloissus Chia
Welcome to Student Manager.
        1. Print students list by ID.
        2. Print students list by Name.
        3. Print student transcript.
        4. Create a new student.
        5. Edit a student.
        6. Delete a student.
        0. Go back to previous menu.
Choice: 4
Please input student name: Linda Yeo
Please choose gender (M/F): f
Error: Student ID 19, Linda Yeo already exists.
```

- Invalid input

```
Please input student name:
Please choose gender (M/F):
Error: Invalid choice.
```

- Print student list by ID

```
Choice: 1
Students List:
NO    | STUDENT ID | NAME
--------------------------------------------------------------------------------
1     | 1          | Kenny Pang
2     | 2          | Lucas Neo
3     | 3          | Tan Yun Hwa
4     | 4          | Loh Jun Kiat
5     | 5          | Duong Canh Vi
6     | 6          | Melissa Teo
7     | 7          | Liu Wan Ling
8     | 8          | Victor Tan
9     | 9          | Duphk Dipth
10    | 10         | Linda Yeo
11    | 11         | Muhammad
12    | 12         | Wendy Koh
13    | 13         | Louis Tan
14    | 14         | Lois Chua
15    | 15         | Sean Ng
16    | 18         | Aloissus Chia
```

- Edit student's particulars

```
Choice: 5
Please input student ID: 11
Student: 11, Muhammad
        General Particulars:
                Name: Muhammad
                Gender: Male
                IC: S0150166N
                Contact: 96885869
                Personal email: S3723@yahoo.com
                Address: #87-7-481 Hall of Residence 6, NTU, Singapore
                Date of Birth: null
        Student Particulars:
                Student Mail: S0715@e.ntu.edu.sg
                School: School of Computer Engineering
                Program: CS
                Student Type: FULL_TIME
                CGPA: 0.0

        1. Edit General Particulars.
        2. Edit Student Particulars.
        3. Edit Student Grades.
        4. Print Student Transcript.
        0. Go back to previous menu.
Choice: 1
Person: 11, Muhammad
        Name: Muhammad
        Gender: Male
        IC: S0150166N
        Contact: 96885869
        Personal email: S3723@yahoo.com
        Address: #87-7-481 Hall of Residence 6, NTU, Singapore
        Date of Birth: null

        1. Edit Name.
        2. Edit Gender.
        3. Edit IC.
        4. Edit Contact.
        5. Edit Personal eMail.
        6. Edit Address.
        7. Edit Date of Birth.
        0. Go back to previous menu.
Choice: 1
Please input new name: Muhammad Aleemudin
Person: 11, Muhammad Aleemudin
        Name: Muhammad Aleemudin
        Gender: Male
        IC: S0150166N
        Contact: 96885869
        Personal email: S3723@yahoo.com
        Address: #87-7-481 Hall of Residence 6, NTU, Singapore
        Date of Birth: null
```

- Delete a student

```
Choice: 6
Please input student ID: 18
Student 18, Aloissus Chia removed.
```

- Similar for creating and managing Professors (with different particulars)

Test case 2: Course

- Create a course (including its type, AU weightage, professor in charge, inclusion of final exam and capacity)

```
Choice: 3
Welcome to Course Manager.
        1. Print courses list by ID.
        2. Print courses list by Name.
        3. Create a new course.
        4. Edit a course.
        5. Delete a course.
        0. Go back to previous menu.
Choice: 3
Please input course name: Discrete Mathematics
Please input academic units: 3
Please choose the following Course Types:
        1. Lectures only
        2. Tutorials only
        3. Labs only
        4. Lectures and tutorials
        5. Lectures, tutorials and labs
        6. Lectures and labs
        7. Tutorials and labs
Choice: 4
Please input ID of professor as Course Coordinator: 4
Does this course constitutes a final exam? (Y/N): y
Please input course capacity: 12
```

- Print course list

```
Choice: 1
Courses List:
NO     | COURSE ID   | COURSE NAME
-------------------------------------------------------------------------------
1      | 1           | Object Oriented Programming
2      | 2           | Algorithms
3      | 3           | Effective Communication
4      | 4           | Discrete Mathematics
```

- Add an existing course

```
Choice: 3
Please input course name: Algorithms
Please input academic units: 3
Please choose the following Course Types:
        1. Lectures only
        2. Tutorials only
        3. Labs only
        4. Lectures and tutorials
        5. Lectures, tutorials and labs
        6. Lectures and labs
        7. Tutorials and labs
Choice: 4
Please input ID of professor as Course Coordinator: 2
Does this course constitutes a final exam? (Y/N): y
Please input course capacity: 12
Error: Course ID 5, Algorithms already exists.
```

- Invalid entries.

```
Choice: 3
Please input course name:
Please input academic units:
1
Please choose the following Course Types:
        1. Lectures only
        2. Tutorials only
        3. Labs only
        4. Lectures and tutorials
        5. Lectures, tutorials and labs
        6. Lectures and labs
        7. Tutorials and labs
Choice: 1
Please input ID of professor as Course Coordinator: 3
Does this course constitutes a final exam? (Y/N): n
Please input course capacity: a
Error: Input Mismatch. Please provide compatible inputs.
NOTICE: Please save your work or progress to avoid loss of data.
Please select a school:
        1. School of Computer Engineering
        0. Save and Quit.
```

- Edit a course details, including entering course assessment weightage

```
Choice: 1
Please input name of assessment: Mid-term examination
Please input weight of assessment: 30.00
Course: 4, Discrete Mathematics
        Course ID: 4
        Course Name: Discrete Mathematics
        Course AU: 3
        Course Type: Lectures and tutorials
        Course Capacity: 12
        Course Weights:
                Final Exam: 100.00
                Mid-term examination: 30.00
                Total Weights: 130.00
        Course Groups:
                Lecture Group ID: 12
                        No. of Students: 0
                Tutorial Groups: 0 group(s)
                        ID    | No. of Students | GROUP NAME
                        ------------------------------------------------------------------------------
                Total no. of students: 0
        Course Coordinator: 4, Kayla Lin

        1. Edit Course Name.
        2. Edit Course AU.
        3. Edit Course Type.
        4. Edit Course Capacity.
        5. Edit Course Assessments Weights.
        6. Edit Course Groups.
        7. Edit Course Coordinator.
        0. Go back to previous menu.
Choice: 5
Please choose the following:
        1. Add or Edit assessment.
        2. Delete assessment.
Choice: 4
Please input Course ID: 4
Course: 4, Discrete Mathematics
        Course ID: 4
        Course Name: Discrete Mathematics
        Course AU: 3
        Course Type: Lectures and tutorials
        Course Capacity: 12
        Course Weights:
                Final Exam: 100.00
                Total Weights: 100.00
        Course Groups:
                Lecture Group ID: 12
                        No. of Students: 0
                Tutorial Groups: 0 group(s)
                        ID    | No. of Students | GROUP NAME
                        ------------------------------------------------------------------------------
                Total no. of students: 0
        Course Coordinator: 4, Kayla Lin

        1. Edit Course Name.
        2. Edit Course AU.
        3. Edit Course Type.
        4. Edit Course Capacity.
        5. Edit Course Assessments Weights.
        6. Edit Course Groups.
        7. Edit Course Coordinator.
        0. Go back to previous menu.
Choice: 5
Please choose the following:
        1. Add or Edit assessment.
        2. Delete assessment.
```

```
Choice: 1
Please input name of assessment: Final Exam
Please input weight of assessment: 70.00
Course: 4, Discrete Mathematics
        Course ID: 4
        Course Name: Discrete Mathematics
        Course AU: 3
        Course Type: Lectures and tutorials
        Course Capacity: 12
        Course Weights:
                Final Exam: 70.00
                Mid-term examination: 30.00
                Total Weights: 100.00
        Course Groups:
                Lecture Group ID: 12
                        No. of Students: 0
                Tutorial Groups: 0 group(s)
                        ID    | No. of Students | GROUP NAME
                        ------------------------------------------------------------------------------------
                Total no. of students: 0
        Course Coordinator: 4, Kayla Lin
```

- 3 components

```
        5. Edit Course Assessments Weights.
        6. Edit Course Groups.
        7. Edit Course Coordinator.
        0. Go back to previous menu.
Choice: 5
Please choose the following:
        1. Add or Edit assessment.
        2. Delete assessment.
Choice: 1
Please input name of assessment: Group Project
Please input weight of assessment: 50
Course: 3, Effective Communication
        Course ID: 3
        Course Name: Effective Communication
        Course AU: 2
        Course Type: Tutorials only
        Course Capacity: 12
        Course Weights:
                Personal Reflection: 20.00
                Group Project: 50.00
                Oral Presentation: 30.00
                Total Weights: 100.00
        Course Groups:
                Tutorial Groups: 1 group(s)
                        ID    | No. of Students | GROUP NAME
                        ------------------------------------------------------------------------------------
                        11    |       0/15      | Group 11
                Total no. of students: 0
        Course Coordinator: 3, Vanissa Leong
```

Test case 3: Semester management

- Add courses to a semester

```
Choice: 1
Welcome to School of Computer Engineering
Please select an option:
        1. Manage Students
        2. Manage Professors
        3. Manage Courses
        4. Manage Semesters
        0. Go back to previous menu
Choice: 4
Semester Registration Manager.
        1. Print courses open for registration.
        2. Add a course to current semester.
        3. Add all courses to current semester.
        4. Remove a course from current semester.
        5. Remove all courses from current semester.
        6. Register a student to a course.
        7. Edit student's group in a course.
        8. Manage grades for a student of a course.
        9. Unregister a student from a course.
        10. Unregister all students from a course.
        11. End current semester.
        0. Go back to previous menu.
Choice: 3
All courses added successfully.
        Courses List:
        NO    | COURSE ID  | COURSE NAME
        ------------------------------------------------------------------------------
        1     | 1          | Object Oriented Programming
        2     | 2          | Algorithms
        3     | 3          | Effective Communication
        4     | 4          | Discrete Mathematics
```

- Register a student to a course, showing course vacancies

```
Choice: 6
Please input student ID: 1
Please input Course ID: 1
Lecture Group ID: 1
        No. of Students: 0
Tutorial Groups: 2 group(s)
        ID    | No. of Students | GROUP NAME
        ------------------------------------------------------------------------------
        3     |       0/6       | Group 3
        4     |       0/6       | Group 4
Lab Groups: 2 group(s)
        ID    | No. of Students | GROUP NAME
        ------------------------------------------------------------------------------
        5     |       0/6       | Group 5
        6     |       0/6       | Group 6
Total no. of students: 0

Please input Lecture group: 1
Please input Tutorial group: 3
Please input Lab group: 5
Student ID 1, Kenny Pang registered in Course ID 1, Object Oriented Programming successfully.
```

- Register a student to a tutorial or lab group with no vacancies

```
Choice: 6
Please input student ID: 7
Please input Course ID: 1
Lecture Group ID: 1
        No. of Students: 6
Tutorial Groups: 2 group(s)
        ID    | No. of Students | GROUP NAME
        ------------------------------------------------------------------------------------
        3     |       6/6       | Group 3
        4     |       0/6       | Group 4
Lab Groups: 2 group(s)
        ID    | No. of Students | GROUP NAME
        ------------------------------------------------------------------------------------
        5     |       6/6       | Group 5
        6     |       0/6       | Group 6
Total no. of students: 6

Please input Lecture group: 1
Please input Tutorial group: 3
Please input Lab group: 5

Error: Liu Wan Ling not enrolled as Group 3 is full..
Please register the student again in another group.
```

- Register for the same course again

```
Choice: 6
Please input student ID: 1
Please input Course ID: 1
Lecture Group ID: 1
        No. of Students: 6
Tutorial Groups: 2 group(s)
        ID    | No. of Students | GROUP NAME
        ------------------------------------------------------------------------------------
        3     |       6/6       | Group 3
        4     |       0/6       | Group 4
Lab Groups: 2 group(s)
        ID    | No. of Students | GROUP NAME
        ------------------------------------------------------------------------------------
        5     |       6/6       | Group 5
        6     |       0/6       | Group 6
Total no. of students: 6

Please input Lecture group: 1
Please input Tutorial group: 4
Please input Lab group: 6
Error: Student Kenny Pang is already enrolled..
```

- Invalid data entries

```
Choice: 6
Please input student ID: 8
Please input Course ID: 5
Error: Course ID 5 does not exists in current semester.


Please input student ID: 1
Please input Course ID: 2
Lecture Group ID: 2
        No. of Students: 0
Tutorial Groups: 2 group(s)
        ID    | No. of Students | GROUP NAME
        ------------------------------------------------------------------------------------
        7     |       0/6       | Group 7
        8     |       0/6       | Group 8
Lab Groups: 2 group(s)
        ID    | No. of Students | GROUP NAME
        ------------------------------------------------------------------------------------
        9     |       0/6       | Group 9
        10    |       0/6       | Group 10
Total no. of students: 0

Please input Lecture group: 0
Please input Tutorial group: 0
Please input Lab group: 0]
Error: Input Mismatch. Please provide compatible inputs.
NOTICE: Please save your work or progress to avoid loss of data.
```

- Add exam and coursework grade

```
Student: 1, Kenny Pang
Course: 1, Object Oriented Programming
        GPA: 0.00
        Overall Marks: 0.00
        Component Marks:
                Final Exam: 0.00 => 0.00
                Assignment: 0.00 => 0.00

1. Final Exam
2. Assignment
0. Go back to previous menu.
Choice: 2
Please input marks: 80.00
Grade Manager.
Student: 1, Kenny Pang
Course: 1, Object Oriented Programming
        GPA: 1.50
        Overall Marks: 24.00
        Component Marks:
                Final Exam: 0.00 => 0.00
                Assignment: 80.00 => 24.00

1. Final Exam
2. Assignment
0. Go back to previous menu.
Choice: 1
Please input marks: 70.00
Grade Manager.
Student: 1, Kenny Pang
Course: 1, Object Oriented Programming
        GPA: 4.00
        Overall Marks: 73.00
        Component Marks:
                Final Exam: 70.00 => 49.00
                Assignment: 80.00 => 24.00
```

- Invalid entries

```
Choice: 8
Please input student ID: 20
Please input Course ID: 1
Error: Student ID 20 does not exist.
Choice: 8
Please input student ID: 1
Please input Course ID: 6
Error: Course ID 6 does not exists in current semester.
Choice: 1
Please input marks: 110
Error: Marks cannot be higher than 100%.
```

- Print students enlisted in a lecture/tutorial/lab group

```
Choice: 11
Please input Course ID: 1
                Lecture ID, Name: 1, Group 1
                Lecture Vacancy: 6 / 12
                Tutorial ID, Name: 3, Group 3
                Tutorial Vacancy: 6 / 6
                        Students list of Group: 3, Group 3
                        NO    | STUDENT ID | NAME
                        ------------------------------------------------------------------------------
                        1     | 1          | Kenny Pang
                        2     | 2          | Lucas Neo
                        3     | 3          | Tan Yun Hwa
                        4     | 4          | Loh Jun Kiat
                        5     | 5          | Duong Canh Vi
                        6     | 6          | Melissa Teo
                Tutorial ID, Name: 4, Group 4
                Tutorial Vacancy: 0 / 6
                        Students list of Group: 4, Group 4
                        NO    | STUDENT ID | NAME
                        ------------------------------------------------------------------------------
                Lab ID, Name: 5, Group 5
                Lab Vacancy: 6 / 6
                        Students list of Group: 5, Group 5
                        NO    | STUDENT ID | NAME
                        ------------------------------------------------------------------------------
                        1     | 1          | Kenny Pang
                        2     | 2          | Lucas Neo
                        3     | 3          | Tan Yun Hwa
                        4     | 4          | Loh Jun Kiat
                        5     | 5          | Duong Canh Vi
                        6     | 6          | Melissa Teo
                Lab ID, Name: 6, Group 6
                Lab Vacancy: 0 / 6
                        Students list of Group: 6, Group 6
                        NO    | STUDENT ID | NAME
                        ------------------------------------------------------------------------------
```

- Invalid entries

```
Choice: 11
Please input Course ID: 4
Error: Course ID 4 does not exists in current semester.
```

- Print course statistic

```
Please input Course ID: 2
        Course Statistics:
        Students:
                1, Kenny Pang:
                        GPA: 3.50
                        Overall Marks: 66.00
                        Component Marks:
                                Final Exam: 60.00 => 42.00
                                Assignment: 80.00 => 24.00
                2, Lucas Neo:
                        GPA: 5.00
                        Overall Marks: 92.00
                        Component Marks:
                                Final Exam: 95.00 => 66.50
                                Assignment: 85.00 => 25.50
                3, Tan Yun Hwa:
                        GPA: 3.50
                        Overall Marks: 69.00
                        Component Marks:
                                Final Exam: 60.00 => 42.00
                                Assignment: 90.00 => 27.00
                4, Loh Jun Kiat:
                        GPA: 4.00
                        Overall Marks: 71.00
                        Component Marks:
                                Final Exam: 80.00 => 56.00
                                Assignment: 50.00 => 15.00
                5, Duong Canh Vi:
                        GPA: 5.00
                        Overall Marks: 93.00
                        Component Marks:
                                Final Exam: 90.00 => 63.00
                                Assignment: 100.00 => 30.00
                6, Melissa Teo:
                        GPA: 3.50
                        Overall Marks: 69.50
                        Component Marks:
                                Final Exam: 65.00 => 45.50
                                Assignment: 80.00 => 24.00
                Overall Mean: 460.50 / 6 = 76.75
```

- Invalid entries

```
Choice: 13
Please input Course ID: 4
Error: Course ID 4 does not exists in current semester.
```

- Print student transcript

```
Choice: 12
Please input Student ID: 5
        Student: 5, Duong Canh Vi
        CGPA: 0.00
        Courses taken:
        Semester: Sem 1
                Course: 1, Object Oriented Programming
                        GPA: 4.50
                        Overall Marks: 80.00
                        Component Marks:
                                Final Exam: 80.00 => 56.00
                                Assignment: 80.00 => 24.00
                Course: 2, Algorithms
                        GPA: 5.00
                        Overall Marks: 93.00
                        Component Marks:
                                Final Exam: 90.00 => 63.00
                                Assignment: 100.00 => 30.00
```

- Invalid entries

```
Choice: 12
Please input Student ID: 18
Error: Student ID 18 does not exist.
```

---END---