LAB211 Assignment

Type:	**Short Assignment**
Code:	**J1.S.P0065**
LOC:	**70**
Slot(s):	**1**

Title

Check data format.

Background

N/A

Program Specifications

Create a program allows input:

- Student information includes: Student name, class name, the marks Math, Physical and Chemistry in the range from 1 to 10.

Display on screen the information:

- Student Type following the conditions:
  - A: mark > 7.5.
  - B: 6 <= mark <= 7.5.
  - C: 4 <= mark < 6 .
  - D: mark < 4.
- Student Type statistics by %.

Function details:

Function 1: Display GUI And Input Data.

- Users run the program. The program prompts users to input Student data.
- When users stop inputting Student data, next **Function 2**.

Function 2: Perform function

- The program classifies students and gives student rank statistics by %.
- Display notify result of students together with statistic result and exit the program.

Expectation of User interface:

1

```
====== Management Student Program ======
Name:Nghia
Classes:FU1
Maths:11
Maths is less than equal ten
Maths:-1
Maths is greater than equal zero
Maths:
Maths is digit
Maths:10
Chemistry:11
Chemistry is less than equal ten
Chemistry:-1
Chemistry is greater than equal zero
Chemistry:
Chemistry is digit
Chemistry:10
Physics:11
Physics is less than equal ten
Physics:-1
Physics is greater than equal zero
Physics:
Physics is digit
Physics:10
Do you want to enter more student information?(Y/N):Y
Name:Nghia 2
Classes:FU1
Maths:10
Chemistry:10
Physics:10
Do you want to enter more student information?(Y/N):N
```

2

```
------ Student1 Info ------
Name:Nghia
Classes:FU1
AVG:10.0
Type:A
------ Student2 Info ------
Name:Nghia 2
Classes:FU1
AVG:10.0
Type:A
--------Classification Info -----
A: 100.0%
B: 0.0%
C: 0.0%
D: 0.0%
```

Guidelines

Student must implement the methods

- createStudent
- averageStudent
- getPercentTypeStudent

in startup code.

Suggestion:

Create a class Student contains the following properties:

- Student Name, Class, Math, Physical, Chemistry, Average, Type.

Create a class Mark Calculation, classify students, calculate Student Type statistics, and set the results on Student object.

Give the statistics:

A:? %

B:? %

C:? %

D:? %

Function1: Input student information

- Must create function: Student createStudent(String name, String classes, double maths, double chemistry, double physics)
    - Input:
        - name: Student name
        - classes: Class.
        - maths: Math mark.
        - chemistry: Chemistry mark.
        - physics: Physical mark.
    - Return:
        - Student object.

Function 2: Student classification.

- Write Function: List<Student> averageStudent(List<Student> students)
    - Input:
        - students: the list of students not classified yet.
    - Return: the list of students already classified.

Function3: Student Type Statistics.

- Must create function: HashMap<String, Double> getPercentTypeStudent(List<Student> students)
    - Input:
        - students: the list of students already classified.
    - Return: Student Type statistics by % by the key A,B,C, D.