

Assumption University
Vincent Mary School of Science and Technology

Online Midterm Exam
Semester 2/2021

Subject: CSX3002 / ITX2001 Object-Oriented Concepts and Programming
IT2371 Object-Oriented Programming I

Section: 541, 542

Date: Tuesday January 18th, 2021

Time: 15:00 - 17:00 (2 hours)

Lecturer: A. Pawut Satitsuksanoh and A. Kiratijuta Bhumichitr

Instructions:

1. You are required to do all questions and submit them to the particular MS Teams.
2. Plagiarism is considered as a serious matter. Zero score will be given and the matter will be reported to the director of the program.
3. There are 3 questions in this exam.

Question 1	5 points
Question 2	5 points
Question 3	<u>10</u> points
Total	<u>20</u> points

Total 3 pages (excluding this page)

Instructions for Question 1:

- Create a package "MQ1-ID" when **ID** is your student identification number.
- Name your source file as "MQ1-ID.java"

1. (5 points) Given a method signature,

```
public static String countRepeatN(String inStr, int n)
```

This method will return an integer which is the number of occurrence of n -repeated characters in the string, `inStr`. Let's say that n -repeated in a string is a character appearing n times in a row, where n is the positive integer.

For examples:

- `countRepeatN("hello", 2) → 1`
- `countRepeatN("123456", 1) → 6`
- `countRepeatN("xxxabyyyycd", 2) → 5`
- `countRepeatN("xxxabyyyycd", 3) → 3`
- `countRepeatN("191233333315aabb", 5) → 2`

Complete the given method and write the complete program to call this method and then show at least 10 different results.

Instructions for Question 2:

- Create a package "MQ2-ID" when **ID** is your student identification number.
- Name your source file as "MQ2-ID.java"

2. (5 points) Given a method signature,

```
public static String removeStars(String inStr)
```

This method will return a version of the given string, where for every star ("*" or "star") in the string, the star and the **chars** immediately to its left and right are gone. So "thisstar" yields "thi" and "aa*ss" also yields "as". You may assume that the string "star" is always in the lowercase.

For examples:

- `removeStars("a*b") → ""`
- `removeStars("a*a*bb*bb") → "b"`
- `removeStars("*test*") → "es"`
- `removeStars("a*staring") → "ng"`
- `removeStars("juststartexam") → "jusexam"`

Complete the given method and write the complete program to call this method and then show at least 10 different results.

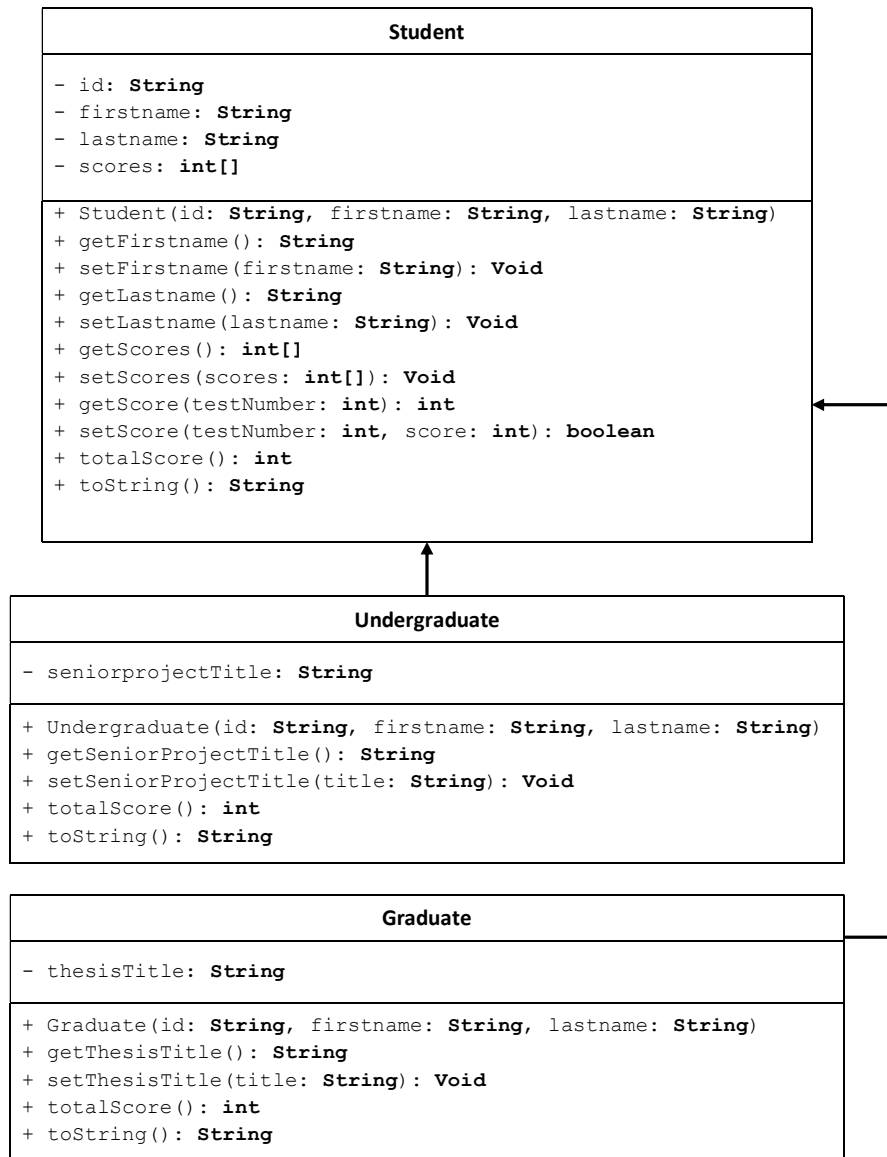
Question 3 is on the next page.

Instructions for Question 3:

- Create a package “MQ3-ID” when *ID* is your student_identification number.
- Name your main program as **StudentTest** and put all class files in “MQ3-ID” package.
- MQ3-ID package must contain **four** .java files, which are **Student.java**, **Undergraduate.java**, **Graduate.java**, and **StudentTest.java**.
- Compress(zip) MQ3-ID package/folder and upload to the particular MS Teams account.

3. (10 points) Student Score System

Consider the given class diagram,

**3.1 Student Class Instructions:**

- Constructor takes 3 parameters, student ID, first name and last name. The size of the score array is 5 tests. The initial score of each test is 0.
- Implement **Getter and Setter** methods for private fields.
- **getScore()** method returns the score of the given test number.

- **setScore()** method updates the score of the given test number and returns true if the update can be successfully made. The method returns false if the given test number and/or the score is invalid then the score will not be updated.
- **toString()** method returns string according to this format "{id} {firstname} {lastname}".

3.2 Undergraduate Class Instructions:

- Constructor takes 3 parameters, student ID, first name and last name. The size of the score array is 5 tests. The initial score of each test is 0.
- The valid range of the score starts from 0 to 5.
- Implement **Getter and Setter** methods for private fields.
- **totalScore()** override method returns the sum of all score and divide by 5.
- **toString()** method returns string according to this format "{id} Undergraduate {totalScore}".

3.3 Graduate Class Instructions:

- Constructor takes 3 parameters, student ID, first name and last name. The size of the score array is 3 tests. The initial score of each test is 0.
- The valid range of the score starts from 0 to 10.
- Implement **Getter and Setter** methods for private fields.
- **totalScore()** override method returns the sum of all score and divide by 3.
- **toString()** method returns string according to this format "{id} Graduate {totalScore}".

3.4 StudentTest class Instructions:

- Implement a class called StudentTest which contains the main() method to test and show that **all of the constructors and methods** work correctly according to the class diagram.

End of the Examination
