CS2043 Assignment 1 — Programming Review

Due: Mon JAN-17 2022 by 4PM

Update (2022 JAN-14): Changed output to match unambiguous and consistent conversion specification

Try this assignment mostly on your own. If you want help, go to the lab on Fri JAN-14. Feel free to talk amongst each other about your solutions, but don't share code with anyone. A quick look at a screen during a conversation is fine. Anything beyond that is cheating.

Even if you don't finish, submit what you get done by the due date. You won't get credit for submitting just anything, but if its worth assessing, you will get credit, even if you don't pass. Note that solutions which are unreasonably similar to other submissions will be reported as an academic offense. Also, solutions which are barely complete, or approached poorly will not be assessed.

Problem Statement:

Create a `Grade` class, a `Transcript` class and a `Registrar` class which do the following, based on well designed object-oriented code. Consider variable type, scope, and modifiers like static and final in your solution. Other methods besides the ones delineated below may be necessary to meet the specifications.

Your system must provide a way to add grades to a transcript, remove grades from a transcript and calculate a GPA based on the grades in a transcript. Provide a remove method that removes all instances of a grade with the same name, and one that removes only instances with the same name and letter grade. Each grade item must include a course number, a grade letter, a grade point and a credit hour, but only 1 grade format needs to be specified on creation; the other should be calculated via static methods in the `Registrar` (a utilities class for grade manipulation). Empty `Grade` items are not allowed. `Transcripts` must have a numeric student Id associated with them, but are allowed to have no assigned grades. The `Transcript` should rely on the `Registrar` to calculate the GPA. Both `Grade` and `Transcript` should override `toString()`, but the `Transcript` should include a `printTranscript()` method to provide a more detailed output of the transcript (see example output).

To demonstrate the functionality of your classes, create a `TranscriptDriver`. The driver should do the following in order:

→ create a transcript with student ID 8510804.
→ Add the following grades to the transcript: CS1003 4CH B, CS2033 4CH A, ECE2213 4CH 3.3, ECON1001 3CH 2.0 CS2033 4CH B
→ print the transcript in the console:

```
Student 8510804: GPA 3.12
CS1003 4CH 3.0 (B)
CS2033 4CH 4.0 (A)
ECE2213 4CH 3.3 (B+)
ECON1001 3CH 2.0 (C)
CS2033 4CH 3.0 (B)
```

For conversion specification, see
https://www.unb.ca/fredericton/engineering/depts/civil/current-undergrad/grades.html

→ Remove the grade with properties CS2033 and B
→ Print the transcript again

→ Add the following grade to the transcript: CS2033 B (ie add the grade that you removed back in)
→ Remove all the grades with the property CS2033
→ Print the transcript again

→ create a transcript with student ID 8510805
→ print the transcript in the console:

```
Student 8510805: GPA  NaN
```

Be sure to submit each of the .java files (not the .class files):

- ☐ **Grade.java**
- ☐ **Transcript.java**
- ☐ **Registrar.java**
- ☐ **TranscriptDriver.java**

We will be marking this assignment against the course coding standards which will be available in D2L under the quick links. We will also assume that you are using the Eclipse IDE to code your assignments, since that is the IDE you will be required to use during the lab exam. For subsequent assignments we will be using GitHub to submit our assignments. For this first assignment however, we will simply submit our code through D2L. Follow the instructions below to submit:

Only solutions submitted in the correct format will be assessed. Your Implementation should be saved in a package labeled **assn1.xyz** (where xyz refer to your initials - if you don't have a middle initial, include it as 'y'). If you set up packaging properly in Eclipse, the package declaration will be automatically generated at the top of each of your java files. If you didn't use Eclipse, be sure to manually declare the package as the first line in each of your files (eg. `package assn1.xyz;`) before you submit.

Submit your files to the folder for Assignment 1 in D2L. **Submit each .java file separately (DO NOT ZIP YOUR PACKAGE)**. You can submit more than once until the assignment deadline.