

Due:

Sunday, 29-November-2019 by 23:59

Deliverables:

The following Java file should be submitted to MS Teams by the due date and time specified above. Submissions received after the deadline will be subject to the late policy described in the syllabus.

- CourseGrade_{StudentNumber}.java

Specifications:

Overview: You will continue the program this week to maintain the grades for a student in a course. Do not forget your headers with @author and @since information. This program will be expanded in future weeks, so be sure you understand the concepts covered in this program.

Requirements: Write a program that will simulate the user interface for a course. It will first prompt the instructor for four (4) categories where each will have:

1. Name – String
 - a. Should be assigned as the value the instructor enters with the first letter capitalized and all other letters lower case
2. Quantity – Integer
 - a. How many of these were given in the course?
 - b. If the value passed is negative, the program will ask again for a valid value
3. Weight – Integer
 - a. What percentage of the grade does this represent?
 - b. If the value passed is negative or makes the total weight of all categories greater than 100, the program will continue to ask for a valid value

Once the instructor has entered the four categories' information, the program will check if the total of all weights is 100. If not, it will display an error message and exit.

Otherwise, the program will then prompt the instructor for points for the individual items in each of the four categories. Note: for this assignment, assume all these decimal values are valid. Also, for simplicity, each item will be considered as out of 100 points.

Once the instructor has completed the data entry, the program will display the student's category percentages, overall grade, and grade letter.

To facilitate the execution of this program, you will write (at minimum) the following methods:

1. main()
 - a. A method to run the CourseGrade program
 - b. It will ask the instructor for the four category information.
 - c. It will then run according to the description above until the instructor has completed entering grades
 - d. Returns None

2. `capitalize(name)`
 - a. Format name with the first letter uppercase; all other letters lowercase
 - b. Takes one String type parameter
 - c. Returns a String
3. `validQuantity(quantity)`
 - a. A method to verify the quantity is valid
 - b. Takes an integer type parameter for the current quantity
 - c. Returns a boolean
 - i. True if quantity is positive
 - ii. False for all other values
4. `validWeight(weight, totalWeight)`
 - a. A method to verify the weight is valid
 - b. Takes two integer type parameters for the weight and total weight so far
 - c. Returns a boolean
 - i. True if weight is non-negative and total weight + weight is not greater than 100
 - ii. False for all other values
5. `gradeLetter(grade)`
 - a. A method to determine the grade letter based on the grade calculated
 - b. Takes one decimal parameter for the grade earned
 - c. Returns a String for the grade letter (refer to assignment 1 for details)
6. `gpaPoints(grade)`
 - a. A method to determine the gpaPoints based on the grade calculated
 - b. Takes one decimal parameter for the grade earned
 - c. Returns a decimal for the GPA points (refer to assignment 1 for details)
7. `status(grade)`
 - a. A method to determine if the student passed based on the grade
 - b. Takes one decimal parameter for the grade earned
 - c. Returns a String containing the status of "passed", "conditionally passed", or "failed" (refer to assignment 1 for details)
8. Any other methods you feel helpful can be implemented, however, these will be the only methods tested.

Design:

When `capitalize` is called with the given parameters, the following values should be returned:

```
▶ CourseGrade_123456789.capitalize("TEST")
Test
▶ CourseGrade_123456789.capitalize("another test")
Another test
```

When `validQuantity` is called with the given parameters, the following values should be returned:

```
▶ CourseGrade_123456789.validQuantity(10)
true
▶ CourseGrade_123456789.validQuantity(0)
false
▶ CourseGrade_123456789.validQuantity(-10)
false
```

When `validWeight` is called with the given parameters, the following values should be returned:

```
▶ CourseGrade_123456789.validWeight(20, 20)
  true
▶ CourseGrade_123456789.validWeight(-20, 20)
  false
▶ CourseGrade_123456789.validWeight(20, 90)
  false
```

When `gradeLetter` is called with the given parameters, the following values should be returned:

```
▶ CourseGrade_123456789.gradeLetter(95)
  AA
▶ CourseGrade_123456789.gradeLetter(65)
  CC
▶ CourseGrade_123456789.gradeLetter(35)
  FD
▶ CourseGrade_123456789.gradeLetter(5)
  FF
```

When `gpaPoints` is called with the given parameters, the following values should be returned:

```
▶ CourseGrade_123456789.gpaPoints(95)
  4.0
▶ CourseGrade_123456789.gpaPoints(65)
  2.0
▶ CourseGrade_123456789.gpaPoints(35)
  0.5
▶ CourseGrade_123456789.gpaPoints(5)
  0.0
```

When `status` is called with the given parameters, the following values should be returned:

```
▶ CourseGrade_123456789.status(95)
  passed
▶ CourseGrade_123456789.status(55)
  conditionally passed
▶ CourseGrade_123456789.status(15)
  failed
```

The program should keep asking for a value until a valid value is entered. If the total weight is not exactly 100, the program should terminate with an error message. Examples shown below:

```
***** Category Information Entry *****

Please enter the name of the category 1: category1
Please enter how many items of type Category1 were given: 1
Please enter the percentage weight of Category1: -10
Please enter the percentage weight of Category1: 0
Please enter the name of the category 2: cATEGORY2
Please enter how many items of type Category2 were given: -1
Please enter how many items of type Category2 were given: 0
Please enter how many items of type Category2 were given: 1
Please enter the percentage weight of Category2: 20
Please enter the name of the category 3: cat3
Please enter how many items of type Cat3 were given: 1
Please enter the percentage weight of Cat3: 95
Please enter the percentage weight of Cat3: 40
Please enter the name of the category 4: cat4
Please enter how many items of type Cat4 were given: 1
Please enter the percentage weight of Cat4: 20

ERROR: The values sum to 80 but should sum to 100.
```

When run, your main program should ask the instructor for the category information, then student grade information. When the instructor has completed all data entry, the grade percentages for each category and final grade information will be displayed.

```
***** Category Information Entry *****

>> Please enter the name of the category 1: quiz
>> Please enter how many items of type Quiz were given: 4
>> Please enter the percentage weight of Quiz: 10
>> Please enter the name of the category 2: HOMEWORK
>> Please enter how many items of type Homework were given: 3
>> Please enter the percentage weight of Homework: 20
>> Please enter the name of the category 3: Midterm
>> Please enter how many items of type Midterm were given: 1
>> Please enter the percentage weight of Midterm: 30
>> Please enter the name of the category 4: final
>> Please enter how many items of type Final were given: 1
>> Please enter the percentage weight of Final: 40

***** Student Grades Entry *****

Please enter values that the student earned for each item:
>> Quiz 1: 60
>> Quiz 2: 70
>> Quiz 3: 80
>> Quiz 4: 90
>> Homework 1: 65
>> Homework 2: 75
>> Homework 3: 70
>> Midterm 1: 89.5
>> Final 1: 75.4

***** Student Results *****

Quiz: 75.0
Homework: 70.0
Midterm: 89.5
Final: 75.4
The student has passed CSE 101 with a score of 78.51, GPA points of 3.0, and a grade letter of BB
```

Code: Create variables for the values entered by the user and assign using the Scanner object. As a user enters valid values update the values of the variables accordingly.

Test: You are responsible for testing your program. It is important to not rely solely on the examples presented in this Project description.

Grading:

MS Teams Submission: If anything is ambiguous, it is your responsibility to ask questions. It is also your responsibility to complete this assignment in a timely manner. E-mails with questions regarding this assignment will likely not be answered if received after 17:00 on the Friday before the due date of the assignment. You can submit multiple times, however, we will only grade the last version that you submitted. Be sure to click the “Turn In” button.

Filename: You must name your java file according to the description above. If your file is not named in this way, your submission for this assignment will not be accepted.

Quiz: There will be a quiz based on this assignment given on 1-December at 12:30. The result of this quiz will be used to determine your grade on this assignment. **Note:** if you do not submit the quiz, your score on this assignment will be 0.