

Due:

Sunday, 12-November-2023 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.

- Assignment2_{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 in this program.

Requirements: Write a program that will simulate the user interface for a course. It will first prompt the instructor for five (5) 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 category 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 five 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 five 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.

For this program, you will write (at minimum) the following methods:

1. main()
 - a. A method to run the program
 - b. It will ask the instructor for the five category information.
 - c. It will then run according to the description above until the instructor has completed entering grades
 - d. Returns None
2. formatCategoryName(name)
 - a. Format name with the first letter uppercase; all other letters lowercase
 - b. Takes one String type parameter
 - c. Returns a String

3. `isQuantityValid(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. `isWeightValid(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 `totalWeight + 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 `formatCategoryName` is called with the given parameters, the following values should be returned:

```
Assignment2_123456789.formatCategoryName("TEST")
Test
Assignment2_123456789.formatCategoryName("another test")
Another test
```

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

```
Assignment2_123456789.isQuantityValid(10)
true
Assignment2_123456789.isQuantityValid(0)
false
Assignment2_123456789.isQuantityValid(-10)
false
```

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

```
Assignment2_123456789.isWeightValid(20, 20)
true
Assignment2_123456789.isWeightValid(-20, 20)
false
Assignment2_123456789.isWeightValid(-20, 90)
false
```

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

```
Assignment2_123456789.gradeLetter(95)
AA
Assignment2_123456789.gradeLetter(9)
FF
Assignment2_123456789.gradeLetter(65.4)
CC
Assignment2_123456789.gradeLetter(35.8)
FD
```

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

```
Assignment2_123456789.gpaPoints(95)
4.0
Assignment2_123456789.gpaPoints(9)
0.0
Assignment2_123456789.gpaPoints(65.4)
2.0
Assignment2_123456789.gpaPoints(35.8)
0.5
```

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

```
Assignment2_123456789.status(95)
passed
Assignment2_123456789.status(65.45)
passed
Assignment2_123456789.status(50.1)
conditionally passed
Assignment2_123456789.status(23)
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
Please enter the name of the category 5: CaTeGoRy5
Please enter how many items of type Category5 were given: 5
Please enter the percentage weight of Category5: 15
ERROR: The values sum to 95 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: Attendance
Please enter how many items of type Attendance were given: 1
Please enter the percentage weight of Attendance: 10
Please enter the name of the category 3: Assignment
Please enter how many items of type Assignment were given: 3
Please enter the percentage weight of Assignment: 20
Please enter the name of the category 4: midterm exam
Please enter how many items of type Midterm exam were given: 1
Please enter the percentage weight of Midterm exam: 20
Please enter the name of the category 5: Final Exam
Please enter how many items of type Final exam were given: 1
Please enter the percentage weight of Final exam: 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
Attendance 1: 65
Assignment 1: 65
Assignment 2: 75
Assignment 3: 70
Midterm exam 1: 89.5
Final exam 1: 75.4

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

Quiz: 75.0
Attendance: 65.0
Assignment: 70.0
Midterm exam: 89.5
Final exam: 75.4
The student has passed CSE 101 with a score of 76.06, 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.