

CSE 101 Programming Assignment 4

Page 1 of 4

Due:

Sunday, 17-January-2021 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 read files for both the category information and student grades for a list of students. It will then write two files to show the results of calculating the student grades.

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

1. countCategory(filename)
 - a. A new method to determine how many categories are in the category list
 - b. Takes a String representing the entire filename as a parameter
 - c. Reads the file and counts the lines in the file
 - d. Returns an integer for the number of lines (categories) in the file
2. getCategory(category, quantity, weight, filename)
 - a. A new method to fill the arrays with the category information
 - b. The file will be read and the values put into the arrays
 - c. The file will have the format of category name, quantity, weight separated by spaces
 - i. Note: the category name will NOT have spaces in it
 - d. It will take four parameters
 - i. String array for names of each category
 - ii. Integer array for quantity of each category
 - iii. Integer array for weight of each category
 - iv. String for the filename
 - e. Returns None
3. writeGrades(student, grade, filename)
 - a. A new method to write an output file with student grade information
 - b. The file will have the format of student name, grade value, grade letter, GPA points, and status separated by spaces
 - c. It will take three (3) parameters
 - i. String array for names of the students

- ii. Decimal array for grade of each student
 - iii. String for the filename
 - d. Returns None
- 4. main(args)
 - a. You will have a main method in this program
 - b. We will pass the base filename as the command line argument
 - c. It will
 - i. Read the category information from a file called {base filename} + "_CourseDetails.txt"
 - ii. Read the student information from a file called {base filename} + "_StudentScores.txt"
 - iii. Write the student grades to a file called {base filename} + "_StudentGrades.txt"
 - iv. Write any errors to a file called {base filename} + "_log.txt"
 - d. Returns none
- 5. Any other methods you feel helpful can be implemented, however, these will be the only methods tested.

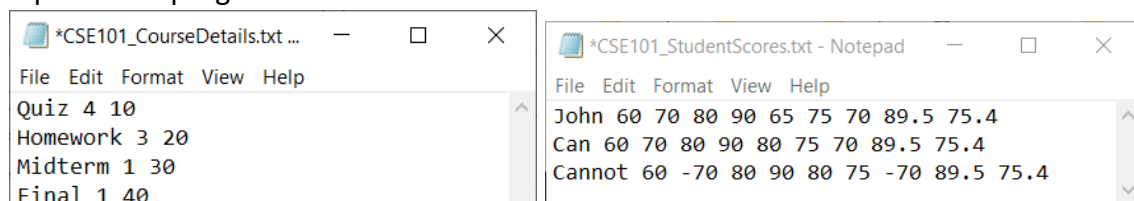
Design:

When your main method is called, your program should display **nothing**. All interaction with the program should be done through reading and writing files.

The expected file formats are as follows:

- 1. _CourseDetails.txt
 - a. Category name, quantity, weight separated by spaces
 - b. Each category on a new line
- 2. _StudentScores.txt
 - a. Student name followed by all scores in order separated by spaces
 - b. Scores are in order based on category and quantity
 - c. Each student on a new line
- 3. _StudentGrades.txt
 - a. Student name, overall score, grade letter, GPA points, status separated by spaces
 - b. Each student on a new line
- 4. _log.txt
 - a. Each error on a new line beginning with "ERROR: "

Code: Prior to running your program, only the following example files could be used as input to the program:



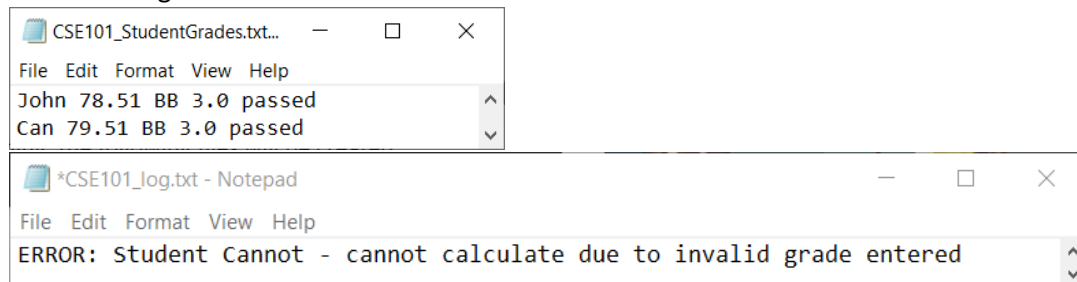
CSE 101 Programming Assignment 4

Page 3 of 4

If we run the following command:

```
java CourseGrade_123456789 CSE101
```

The following files should result:

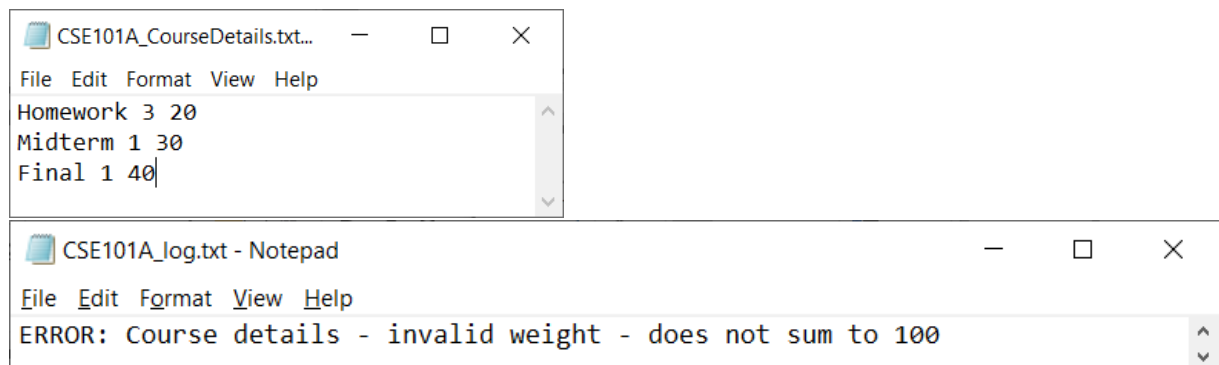


Additionally, your program must check that the category information is entered correctly. If it is not, then the program should write that to the error log and NOT calculate any student scores.

Example below:

If we run the following command:

```
java CourseGrade_123456789 CSE101A
```



NOTE: The error messages displayed above are examples. There are other error conditions (such as a negative weight or quantity) that must also be checked. Make sure that the line in the log starts with the text "ERROR: " and then explains the error condition.

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.

CSE 101 Programming Assignment 4

Page 4 of 4

Exam: Because this assignment is due after the last lecture day, it will not count toward your assignment grade. We will use your score on this assignment as a portion of your final exam grade. Expect that questions related to this assignment will be presented on the final exam.