

Assignment 1

Before attempting this project, be sure you have completed all of the reading assignments, nongraded exercises, discussions, and assignments to date.

Write a Java program which:

- (1) Prompts a user to enter student id, current class grade in GPA format (e.g. 3.5), current class number of credits, overall GPA, and overall number of credits (use Scanner for input).
- (2) Calculates student's new overall GPA based
- (3) Displays the input data along with new GPA to the console

To calculate new overall GPA: ((current-class-GPA times current-class-credits) plus (overallGPA times overall-credits)) divided by (current-class-credits + overall-credits)

- **Example 1:** For this example, given current class 3.5 with 3 credits, and the overall GPA is 3.0 and overall credits is 9 credits: $((3.5*3) + (3.0*9)) / (3+9) = 3.125$ new overall GPA
- **Example 2:** For this example, given current class 3.5 with 3 credits, and the overall GPA is 3.5 and overall credits is 3 credits: $((3.5*3) + (3.5*3)) / (3+3) = 3.5$ new overall GPA
- **Example 3:** For this example, given current class 3.5 with 3 credits and the overall GPA is 4.0 and overall credits is 12 credits: $((3.5*3) + (4.0*12)) / (3+23) = 3.9$ new overall GPA

Summary of the Examples:

<u>Example #</u>	<u>Current Class GPA</u>	<u>Current Class Credits</u>	<u>Overall GPA</u>	<u>Overall Credits</u>	<u>New Overall GPA</u>
1	3.5	3	3	9	3.125
2	3.5	3	3.5	3	3.5
3	3.5	3	4	12	3.9

Test program:

A minimum of 3 test cases should be supplied in the form of table with columns indicating the input values, expected output, actual output and if the test case passed or failed. This table should contain 4 columns with appropriate labels and a row for each test case. An example template is shown below. Note that the actual output should be the actual results you receive when running your program and applying the input for the test record.

Make sure your Java program is using the recommended style such as:

- Javadoc comment up front with your name as author, date, and brief purpose of the program
- Comments for variables and blocks of code to describe major functionality
- Meaningful variable names and prompts
- Class names are written in upper CamelCase
- Constants are written in All Capitals
- Use proper spacing and empty lines to make code human readable

Capture execution:

You should capture and label screen captures associated with compiling your code and running each of your 3 test cases.

Here is a sample run:

```
Enter student id: 6543
Enter current class grade in GPA format (e.g. 3.5): 3.5
Enter current class number of credits: 3
Enter overall GPA: 4.0
Enter overall number of credits: 12
```

STUDENT DATA:

```
Student id: 6543
Current class GPA: 3.5
Current class credits:3
Overall GPA: 4.0
Overall credits: 12
```

NEW GPA: 3.9

Example test cases:

Input	Expected Output	Actual Output	Pass?
id: 6543 class GPA:3.5 class credits: 3 overall GPA: 4.0 overall credits: 12	STUDENT DATA: Student id: 6543 Current class GPA: 3.5 Current class credits: 3 Overall GPA: 4.0 Overall credits: 12 NEW GPA: 3.9	STUDENT DATA: Student id: 6543 Current class GPA: 3.5 Current class credits: 3 Overall GPA: 4.0 Overall credits: 12 NEW GPA: 3.9	Yes
Test Case 2	To be completed	To be completed	

Test Case 3	To be completed	To be completed	
--------------------	-----------------	-----------------	--

Submission requirements

Deliverables include Java program (.java) and a single Word (or PDF) document. The Java and Word/PDF files should be named appropriately for the assignment (as indicated in the SubmissionRequirements document.

The word (or PDF) document should include screen captures showing the successful compiling and running of each of the test cases. Each screen capture should be properly labeled clearly indicated what the screen capture represents. The test cases table should be included in your Word or PDF document and properly labeled as well.

Submit your files to Assignment 1 submission area no later than the due date listed in your online classroom.

Grading Rubric:

The following grading rubric will be used to determine your grade:

Attribute	Level (15-20 points)	Level (5-15 points)	Level 0 (0 - 5 points)
User input	Correct or almost correct prompts and captured input	Mistakes in prompts and/or capture of input	Missing or close to missing user input
Calculation	Correct or almost correct calculation	Mistakes in calculations	Missing or significantly incorrect calculation
Application output	Correct or almost correct output	Mistakes in output data or format	Missing or significantly incorrect output
Test Cases	Correct or almost correct test cases and test execution	Mistakes or incomplete test cases and execution	Missing or significantly incorrect test cases
Program documentation and style	Correct or almost correct program comments, identifiers, and screen captures	Mistakes or incomplete documentation and/or style	Missing or significantly incorrect documentation and/or style