

Computer Science 111 Intro to Algorithms & Programming Lab

Programming Project #3 – Grade Book (30 points)

Due 10/18/18

Create a command-line Java program to input and track student grades. The key to developing this program will be a **sentinel loop** that will allow the user to input student scores, one at a time, until the sentinel value of -1 (or any negative number) is entered, to indicate that the loop should exit and the results should be output. The user will enter a series of **double** numbers which represent test scores between 0 and 100, where an **A** is greater than or equal to 90.0 and less than or equal to 100.0, a **B** is greater than or equal to 80.0 and less than 90.0, a **C** is greater than or equal to 70.0 and less than 80.0, a **D** is greater than or equal to 60.0 and less than 70.0 and an **F** is greater than or equal to 0 and less than 60.0. As each **score** is entered, count the **number of A's, B's, C's, D's and F's** earned so that you can **list grade totals after the loop exits**. You will also need to keep track of the **total number of scores entered** as well as the **sum of all the scores entered** so you can **compute the average score after the loop completes**.

To make your code more readable and easier to understand, be sure to use variable names which “make sense”. For example if declaring a variable which will represent a test **score** that the user has entered, name the variable **score** or **testScore**, instead of something nonsensical like num. If declaring a variable which will be used to **count** the total number of scores entered, name the variable something like **count** or **numberOfScores** instead of something nonsensical like times. If declaring a variable which will be used to keep track of the **sum** of all scores entered, name the variable **sum** or **sumOfScores** instead of something nonsensical like t.

While the user is entering scores, output a message that displays the score entered by the user as well as the letter grade the score represents, something like “The score of 72.5 is a C”. The program must output a message like this for every score entered. Then, **AFTER** the user enters a negative score and the program exits the loop, output a report showing all of the following statistics:

```
Grade Report
-----
Total number of A's:  V
Total number of B's:  W
Total number of C's:  X
Total number of D's:  Y
Total number of F's:  Z
Total number of scores input:  XX
Average class score:  YY.YY
Average class grade:  A or B or ...
```

It should be obvious that the total number of scores input (XX) should match the totals of A's + B's + C's + D's + F's.

Be sure to compute the average class score (YY.YY) as a **double** data type value. If the average class score is 82.45, I DO NOT want the average class score to be rounded to 82 or 83.0. In other words, the average class score should have AT LEAST two decimal place precision. More than two decimal places of precision is fine. As for the average class grade, use the average class score: when the average class score is 82.5, that means the average class grade is a **B**.

Break large problems into smaller pieces!

Sometimes it is useful for beginners to complete a smaller, simpler program before tackling the full project. Start with an easy program that only completes part of the project. Name your project **Project3** and your main class **GradeBook**. Always use class, variable and project names that clearly describe what you are creating.

Start Here:

Write a program which allows a user to input an average class score. The program should use some sort of if-else or switch selection to display the average class grade depending on the average class score. The program should only compute one average class grade per run.

Add a Loop:

Add code to the beginning of your program, before the average grade selection, that loops and allows a user to input as many scores as he or she needs to. Within the loop, as each score is entered, the program should determine whether that score is an A, B, C, D or an F. If the score is an A, B, C, D, or F, the program should update the variables which are used to track the program's totals and print the grade message to the console. The user should enter a sentinel value of -1 to exit the loop. Remember to check the input so that the sentinel value is not "counted" as a score (i.e. if the score is less than zero, do not count the score as an F).

Dividing a larger problem into smaller sub-problems is an important technique in computer programming known as the "Divide and Conquer" approach to problem solving.

Please Design First

It is good practice to design your software before you write the code. Here are some design tasks you should complete before you start:

- Write out a list of variables you think you will need in the program (i.e. int totalA; // to track the number of scores that are A's, etc.)
- Flow chart (or pseudocode) the selection structures needed to determine whether a score is an A, B, C... The flow chart for such a selection would look similar to Figure 3.4 on page 82 of the textbook.
- Flow chart (or pseudocode) the loop, include the check for the sentinel value input and update of variables.

In other words, have some idea about HOW you are going to solve the problem BEFORE you write the code which attempts to solve the problem.

When you have completed this exercise, ZIP the entire project folder and upload this ZIP file to Canvas. In NetBeans, click **File** then select **Export Project**, click **To ZIP...**, select the location where you want the ZIP file to be saved, type the name of the ZIP file (remember to include the .zip extension) then click the **Save** and **Export** buttons.

If the IDE you are using does not have an export-project-to-zip option, you may have to manually ZIP the file by navigating to the project folder in a file browser and selecting your OS's "compress folder" option. Under Mac OS, the file browser is called **Finder** which offers a **Compress Folder** option. Under Windows, the file browser is called **Windows Explorer** and it offers a **Send to => Compressed (zipped folder)** option.