In Major GPA Calculator – Assignment02

In part one of this assignment, students will complete code that calculates an average of up to ten numbers. In part two of this assignment, students will use similar logic to construct an "In Major" GPA calculator.

Instructions

1. Download assignment02.zip and put your assignment02 folder into your MIS2402 workspace.
2. Start by editing avg10.html – this calculator will produce an average of up to 10 numbers.

Part 1 – Average Calculator

Below is the formula for calculating an average.

$$\overline{X} = \frac{\sum X}{N}$$

As expressed above, calculating an average involves two steps.

- Add up the numeric values of X you have.
- Divide the total value by N which is the quantity of numeric values you have.

To put it another way, to calculate an average, you need to determine two numbers: the numerator (the sum of X), and the denominator (N).

3. A lot of work has been done for you in avg10.html. The click event handler is complete, you don't need to edit anything there. Take a moment to review the contents of avg10.html file in VS Code.
    a. Notice that the click event handler uses a JavaScript feature called "trim". The "trim" method will remove leading and trailing spaces from a string. So:
        i. " " becomes a zero-length string ""
        ii. " abc " (with a space on the end) becomes "abc"
        iii. The sentence " I type with extra spaces " becomes "I type with extra spaces"
4. Your objective is to complete the getAvg10 function. Here are some hints:
    a. You should create (at least) two variables, one to hold the numerator, and one to hold the denominator of your calculation.
    b. The parameters v1, v2, v3, v4. v5, v6, v7, v8, v9, v10 are there for you to use.
    c. You should error trap these values, and make sure that they are numeric. Notice that these are all valid numbers: 0, 1, -12, 3.14, -19.99. If any of the inputs are not numeric, you should return "Bad data. Try again."
    d. You should count how many of these parameters are not blank.

        To be clear, if the user only provides the values 1110 and 2010 the answer is ***not*** 312. That is: (1110 + 2010) / **10** = 312.

        The correct answer would be 1560. Because (1110 + 2010) / **2** = 1560.

e. When you add up your numeric values, use the parseFloat function to cover the value from a string (as it was retrieved from the HTML form) to a number.

   NOTE: The parseFloat function works like the parseInt function you have seen previously. While the parseInt function will accept a string and return an integer, the parseFloat function will accept a string and return a number with decimal places.

   To get a feel for how parseFloat works, try it out in the Web Developer console. Here's an example:

```
> parseInt('2.14')
<· 2
> parseFloat('2.14')
<· 2.14
> parseInt('spider')
<· NaN
> parseFloat('spider')
<· NaN
```

f. If all of the parameters are blank, then your code will attempt to divide by zero. That's bad. So, in that special case, your code should return "Bad data. Try again."

g. Assuming that good data has been provided, you must return your average, rounded to 2 decimal places (using the toFixed method)

5. The following video has been provided to illustrate a working solution.

   https://misdemo.temple.edu/classexamples/avg10demo.gif

Part 2 – In Major GPA calculator

Your In-Major GPA is a measure of your average scores only in the courses which contribute to your major college degree. Unlike your cumulative GPA, which represents all the grades you received in all college courses, an In-Major GPA represents only the grades you received in the major you declared.

An In-Major GPA is sometimes important for employment and graduate school applications. An in-major GPA may also determine a student's eligibility to graduate with honors recognition.

6. Examine the code found in majorgpa.html. Note the use of the html <select> and <option> tags.

   When a user selects the option for the letter grade "A" the selection will have a value of "4". When a user selects the option for the letter grade "B+" the selection will have a numeric value of "3.3". etc. etc.

7. As before, if a user provides invalid / incorrect data, your calcGPA function should return "Bad data. Try again."

8. As before, if a user provides good input, calculate the In-Major GPA on a 4.0 scale, rounded to two decimal places. A student should be able to determine their In-Major GPA even if they have not yet completed all classes.

CONTINUED

9. As before, a sample solution is shown with the following videos.

   https://misdemo.temple.edu/classexamples/gpa1.gif

   https://misdemo.temple.edu/classexamples/gpa2.gif

   https://misdemo.temple.edu/classexamples/gpa3.gif

When you are done…

10. Be sure to upload both avg10.html and majorgpa.html to assignment 2 on canvas.

How will this assignment be graded?

Your solution will be graded on the percentage of test cases passed. Check the following hints below.

- The functions calcGPA and getAvg10 will be inspected.
- The functions should return correct values, rounded to two decimal places, when provided with good data.
- Bad data should be managed as described in the assignment.
- Incomplete data should be managed as described in the assignment.
- Did you remember to use parseFloat? If you didn't and you are getting weird answers, that might be your problem.
- Don't forget to test getAvg10 for negative numbers and zero.