Project 3

The goal of this project is to test your ability to use the knowledge of Java that you have obtained up to this point in the course.
The focus of the project is

- decision structures
- looping structures
- and methods

Instructions

For this project you will be writing seven (7) methods using a variety of return types and parameters. The program calculates and displays the average and letter grade of a user based on the test scores entered.

Program Functionality

➢ Details

1. Write a method called **displayWelcomeMessage** which will display a welcome message. The welcome message should include your name and the name of the assignment (Project 3). It should be formatted neatly. It should let the user know what the purpose of the program is. This method will return nothing.
2. Write a method called **displayEndingMessage** which will display an ending message to the user who has finished using your program. This should simply be a "Thank you for using this program" type message. It should be formatted neatly. This method will return nothing.
3. Write a method called **pressEnterToContinue** which will pause the screen and display a message saying "Press Enter To Continue…" This method will return nothing.
4. Write a method called **getTestScore** which will prompt the user for an integer test score and return a test score. The test score should be between 0-100, and it should keep asking a new value if the input is not valid
5. Write a method called **calcAverage** which accept as parameters three (3) integer test scores and return the average of the three (3) scores.
6. Write a method called **findLetterGrade** which will accept a double parameter representing a course average and return a character letter grade.
    - A is 90 and above
    - B is 80 and above
    - C is 70 and above
    - D is 60 and above
    - F is below 60
7. Write a method called **calcGPA** which will accept a character parameter representing a letter grade and will return the GPA
    - A is 4.0

- B is 3.0
- C is 2.0
- D is 1.0
- F is NO CREDIT (0.0)

8. Write a method called **runAgain** which will ask the user if they would like to run the program again. It should accept only Y, y, N or n as valid input. The method should not return a value until the user has entered a valid input. The method must return an uppercase Y or N (although the user may have entered it as a lowercase y or n).

➢ Program Steps

Your **main** method will:

1. Display a welcome message to the user
2. Call the **pressEnterToContinue** method
3. Prompt the user for three (3) test scores
4. Calculate & display the average of the three (3) test scores
5. Find & display the letter grade associated with the average
6. Calculate & display the GPA associated with the letter grade
7. Call the **pressEnterToContinue** method
8. Ask the user if they wish to run the program again, determining whether or not to repeat steps 2-7. As long as they say Y, the program should repeat. If they say N, the program should display the ending message.
9. When the user is done, display the ending message to the user and call the **pressEnterToContinue** method

➢ Additional Notes

- Averages should be displayed to two decimal places
- GPA's should be displayed to three decimal places
- Prompts should leave the cursor on the same line when accepting input. Prompts should be descriptive about what the user should be entering to reduce the chance of incorrect input.
- All output should be neat, readable, well-aligned, labeled and have correct spelling/grammar
- Every method MUST have a method header of comments.

Project to do list:

- **Step 1. Requirements**
  - Make sure you completely understand the requirements and ask questions if you need clarification.
  - Make sure that you pay close attention to detail and follow the instructions very carefully ☺
- **Step 2. Design**
  - Write **UML statements** for every method
  - Write **the algorithm/flowchart** for every method
  - Write **the algorithm/flowchart** for main method
  - Put all the algorithms in a word document named **Project3Design.docx**.
    At the very top of the document type YOUR name and CSCI-1250-xxx (where xxx is your section number) on the right-hand side.
  - A sample of the UML entry/algorithm would look like this…

    Method: **sumNumbers**

    UML Entry: **+sumNumbers(num: int, num2:int) : int**

    Algorithm:

    1. Accepts two integer numbers in the parameter list
    2. Calculate the sum by adding the two numbers together
    3. Return the sum

  - Remember: your algorithm is your road map – you will follow it when writing your code. DO NOT WRITE YOUR CODE FIRST AND THEN YOUR ALGORITHM/FLOWCHART! (It should NOT be the last thing that you do.)
- **Step 3. Implementation**
  - Create a **class called Project3**.

  Using the algorithms written:

  - Write the **displayWelcomeMessage** method
  - Write the **displayEndingMessage** method
  - Write the **pressEnterToContinue** method
  - Write the **getTestScore** method
  - Write the **calcAverage** method
  - Write the **findLetterGrade** method
  - Write the **calcGPA** method
  - Write the **runAgain** method
  - Use your algorithm as a starting point for your comments throughout your program
  - Write your program one step at a time, i.e. make sure one thing works before going on to something else.

- Complete the documentation of the application by inserting comments and adhering to programming standards
  - 1250 Coding Standards posted under Course Content in D2L
  - Pay particular attention to indenting, no word-wrapping when printed and comments
  - All methods MUST have a method header of comments

Step 4. Testing

- Test each method and make sure it works as it should
- Test your program thoroughly to make sure your values are correct

Step 5. Delivery

- Create a folder named **LastnameFirstnameProject3** (replace Lastname and Firstname with your own last and first name)
- **Create the javadocs** of your program**.** This is a simple as running the following command at the command prompt: ***javadoc Project3.java***
  You should place the resulting javadoc files (html and related files) in a folder named **javadocs**
- Create three (3) screenshots of your program running and print them
- Place the completed java source code (**Project3.java**), design document (**Project3Design.docx**), **javadocs** folder and screenshots in the folder created (**LastnameFirstnameProject3)**
- Zip up the folder containing your source code, design document, javadocs and screenshots. It should be named **LastnameFirstnameProject3.zip** (again, your first and last name will be substituted here)
- Drop the **zipped folder** into the dropbox in D2L.
- This must be completed and submitted to the dropbox by the due date/time. All projects turned in after the due date will be penalized 10% each day late up to max three days.
- If you have not completed the project, submit whatever you have. Partial credit is better than no credit at all.