CSE 1384 - Vectors and Files
Lab 3

Objectives:

- Continue practicing past concepts
- Practice using files
- Practice using vectors

Assignment:

You will be given a base file of **`main.cpp`** as a starting point. Please use this and *do not change any of the code already present.*

Create a program that will average quiz scores. You'll get initial scores from a file. A testing file (**`grades.txt`**) has been provided for you. The file should always have one grade per line, with the grade allowed to be a floating point number between 0 and 100 (inclusive).

First, you should open the file (with a validation loop – continue looping until the user has given a file that exists). Then, read the content of the file, adding each number to a floating point vector named grades.

Once done, close the file and begin a menu (that should loop until the user wants to stop). The initial menuing has been provided for you (along with some other items and comments to guide you). Your menu options are:

- Exit
    - Stop looping
- Display average
    - Calculates and displays the average of the grades present in the grades vector
- Display grades
    - Displays all grades in the grade vector
    - Accompany this with "Grade <number>:" where the number is the appropriate number. Start this numbering at 1 (as that's what most users expect to see)
- Add new grade
    - Ask for a new grade
    - Validate that this grade is in the valid range (0.0-100.0)
        - Loop until it is
    - Add it to you vector
- Additionally:
    - Be able to handle an incorrect menu option

BONUS / Honors Student Credit:

If you're an honors student, you ***must*** complete this portion. If you do not, it will be counted against you.

If you're not an honors student, you may complete this section for an additional 5 bonus points.

For the final average recorded when exiting the program, determine what letter grade they would receive if that average was recorded. You can use a basic 10-point scale:

- `100-90: A`
- `89-80: B`
- `79-70: C`
- `69-60: D`
- `59-0: F`

Example Execution (without Honors / BONUS):

```
Welcome to quiz grade averaging!

Enter the file name: grades
Error opening. Try again.

Enter the file name: grades.txt

Menu:
0. Exit
1. Display average
2. Display grades
3. Add new grade
Enter choice: 2

Grade 1: 90
Grade 2: 80
Grade 3: 70
Grade 4: 60

Menu:
0. Exit
1. Display average
2. Display grades
3. Add new grade
Enter choice: 3

Enter a grade to add (0.0-100.0): -1
Invalid grade. Please try again.

Enter a grade to add (0-100): 50

Menu:
0. Exit
1. Display average
2. Display grades
3. Add new grade
Enter choice: 2

Grade 1: 90
Grade 2: 80
Grade 3: 70
Grade 4: 60
Grade 5: 50

Menu:
0. Exit
1. Display average
2. Display grades
3. Add new grade
Enter choice: 1

Your average quiz score is: 70

Menu:
0. Exit
1. Display average
2. Display grades
3. Add new grade
Enter choice: 0

Good-bye!
```

Example Execution (WITH Honors / BONUS):

```
Welcome to quiz grade averaging!

Enter the file name: grades.txt

Menu:
0. Exit
1. Display average
2. Display grades
3. Add new grade
Enter choice: 2

Grade 1: 90
Grade 2: 80
Grade 3: 70
Grade 4: 60

Menu:
0. Exit
1. Display average
2. Display grades
3. Add new grade
Enter choice: 3

Enter a grade to add (0.0-100.0): 60

Menu:
0. Exit
1. Display average
2. Display grades
3. Add new grade
Enter choice: 2

Grade 1: 90
Grade 2: 80
Grade 3: 70
Grade 4: 60
Grade 5: 60

Menu:
0. Exit
1. Display average
2. Display grades
3. Add new grade
Enter choice: 1

Your average quiz score is: 72

Menu:
0. Exit
1. Display average
2. Display grades
3. Add new grade
Enter choice: 0


With the average of 72 you'd have a grade of a C.
Good-bye!
```

Deliverables:

- C++ code (.cpp file)
- A document (.pdf) with at least two screenshots showing the program running
    - The two program screenshots should have completely different inputs from each other (I suggest making a different text file than just the one provided as well)
    - The two screenshots must be *legible* to count (too small or pixelated text will not be interpreted)
    - Show *all* possible error messages

Point Breakdown:

(100 points total)

A submission that doesn't contain any code will receive a 0.

- 25pts - files
    - 10pts - properly opens/closes files (5pts each)
    - 10pts - verifies file exists in a loop
    - 5pts - properly reads from the file
- 25pts - vectors
    - 5pts - properly creates a floating point vector
    - 10pts - properly adds to the vector
        - 5pts each: adding items from the read loop, adding items in the menu
    - 10pts - properly displays vector items
- 30pts - menu
    - 10pts - validates number range for adding an item to the vector (in a loop)
        - 0.0-100.0
    - 10pts - each menu option handles what it should:
        - Display average, display grades, add new item (4pts for one right, 3 pts for each additional for 10pts total)
    - 5pts - handles incorrect menu option correctly
    - 5pts - average is properly calculated
- 10pts - two unique screenshots
- 10pts - programming style *
- Honors / BONUS (dependent on Honors status):
    - Honors: 5pts penalty for not completing
    - BONUS: 5pts added for completing

* Programming style includes good commenting, variable nomenclature, good whitespace, etc.