CMPS 2010 Coding Final

Fall 2024

Option 1

TVShow.h, TVShow.cpp (Requirements)

- Declare & Define a class called TVShow with the following:
  - Class Variables:
    - **`string title`**
    - **`string platform`**
    - **`double score`**
  - Constructors:
    - A default constructor that sets all strings to empty strings and the score to 0
  - Class Functions:
    - Setters for all three class variables
    - **`string toString()`** that returns a string using the following format:
      Breaking Bad, Netflix, 9.8

Main.cpp (Requirements)

- Main.cpp will contain two functions:
  - int main()
    - Display a nice greeting to the user and ask how many shows they would like to review
    - Dynamically create a TVShow array using the size provided by the user
    - In a loop (for each TVShow in the array):
      - Prompt the user for the title, platform, and score.
      - Use the setter functions to set the array object values accordingly
    - Call the outputShows function and pass the completed TVShows array and size
  - void outputShows(TVShow shows[], int size)
    - In a loop (for each show in shows[]) :
      - Call the **`toString()`** class function and write the returned string to the screen
    - Close the file

The output of the **`outputShows`** function should look something like this:

```
Breaking Bad,Netflix,9.800000
Always Sunny,Hulu,7.200000
Game of Thrones,HBO,8.600000
```

NOTES:

- The strings provided by the user may contain spaces, so you will probably want to use **getline()**
- For this assignment **you DO NOT need to validate any of the user inputs**.
- To compile your code:

```
g++ Main.cpp TVShow.cpp -o main
```

Option 2

Implement a C++ program that helps users track their personal expenses.
Your program will allow users to input, view, and analyze their spending data.

Finance.cpp (Requirements)

- **User Menu**
  Implement a menu system that provides the following options:
  - Add a new expense.
  - View all expenses.
  - View total expenses and calculate the average expense amount.
  - Exit the program and write the list of expenses to a file.

- **Input and Output:**
  - Allow the user to enter each expense as a decimal.
  - Display all recorded expenses with two decimal points of precision

- **Functions:**
  - Implement functions for each menu option (e.g., addExpense, viewExpenses, calculateTotals).

- **Data Storage:**
  - Use an array to store the expenses during program execution.
  - When the user decides to exit, loop through the array of expenses and write them all to a file.
  - Also write the total and average to the file.

- **Error Handling:**
  - Validate user input (e.g., ensure the expense amount is a positive number).
  - Handle file reading/writing errors gracefully.

- **Program Flow:**
  - The program should loop back to the main menu after each action unless the user chooses to exit.

NOTES:

- To compile your code:

```
g++ Finance.cpp -o main
```

Run this command from your midterm folder to submit your work:

```
/home/fac/paul/s/final.sh
```