

CMPS 2010 Coding Final

Fall 2022

Option 1

Movie.h, Movie.cpp

- Declare & Define a class called Movie with the following:
 - a. Class Variables:
 - `string title`
 - `string genre`
 - `double score`
 - b. Constructors:
 - A default constructor that sets all strings to empty strings and all number values to 0
 - c. Class Functions:
 - Setters for all five class variables
 - `string toCSV()` that returns a string using the following format:
Back To The Future,Comedy,9.8

Main.cpp

- Main.cpp will contain two functions:
 - a. `int main()`
 - Display a nice greeting to the user and ask how many movies they would like to review
 - Dynamically create a Movie array using the size provided by the user
 - In a loop (for each Movie in the array):
 - Prompt the user for the title, genre, and score.
 - Use the setter functions to set the array object values accordingly
 - Call the `outputMovies` function and pass the completed Movies array and size
 - b. `void outputMovies(Movie movies[], int size)`
 - Open a file called "movieRatings.csv"
 - In a loop (for each movie in `movies[]`):
 - Call the `toCSV()` class function and write the returned string to the file
 - Close the file

The contents of the `movieRatings.csv` file should look something like this:

```
Teenage Mutant Ninja Turtles,Action,8.500000
DUNE,Drama,9.500000
Snatch,Comedy,10.000000
```

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 Movie.cpp -o main`

Option 2

Atm.cpp

- Declare & define the following functions:
 - a. `void withdraw(double &balance)`
This will request an amount from the user, validate the input, then subtract the amount from **balance**.
 - b. `void deposit(double &balance)`
This function will request an amount from the user, then add the amount to **balance**.
 - c. `void checkBalance(double balance)`
This function will display the current **balance** to the screen.
The balance should always be displayed with TWO digits after the decimal since it represents money.
- The main function will do the following:
 1. Create a constant int with a secret pin and assign it the value **12345**.
 2. Prompt the user for their pin and use a validation loop to make sure:
 - The pin is a valid integer
 - The pin matches the same value as the SECRET PIN
 3. Once the user has entered a valid pin, you will do the following in a loop:
 - Display a MENU to the user with the following options:
 1. Withdraw
 2. Deposit
 3. Check Balance
 4. Exit
 - Prompt the user for their option and validate the input.
 - For options 1-3 you will call the corresponding function.
 - For option 4 you will display a thank you message and end the loop
 4. Spend some time making sure your menus and prompts are well formatted and easy to follow:

Something like this:

```
Please enter your pin: 12345

#####
## Welcome to PABLO'S CREDIT UNION! ##
#####
## 1. Withdraw ##
## 2. Deposit ##
## 3. Show Balance ##
## 4. Exit ##
#####

Please choose an option: 3
| CURRENT BALANCE: 0.00
```

NOTE:

- Think about how many variables you will need to track the user's pin, menu option, balance, etc.

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

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