Zachary Meisner

12/2/2021

CS210

Prologue

To quickly talk about the development of this program, this probably has been one of the more challenging projects I have had to make so far. I studied all about classes and objects in week 7, and when I felt confident enough to make the program, I decided to think about everything I had to do for the project, and really started to understand the importance of compartmentalizing the different functions of the program, and making the data private for the user's protection, in addition to simplifying the program which after understanding how it works, I can see makes everything much easier to understand.

1. UserData.h

Breaking down the project I had to start with the class file, as it was what was most challenging for me so I started by making the UserData.h file

UserData.h

Create the class

Create public variables

- Getters
- Setters
- Declare functions for display and output
- Declare functions for getting input and testing the input

Create private variables

- Only the variables taken from the user as input

2. UserData.cpp

This is the backend of the UserData header file which is used specifically for the UserData, this was most important to make because without this I could not create the rest of the program.

Set initial investment, Get Initial Investment

Set month deposit, Get Monthly Deposit

Set annual Interest, Get Annual Interest

Set number years, Get number of years

3. Display.cpp

-Display-

Given that this is the first thing the user sees in the program it makes most sense to make this next

Take Display function using the variables that have just been initialized in step 2

Create ASCII display art to make the output look cool

-Get Input-

We again are creating a GetInput function to get input from the user using the variables we just initialized in step 2

This step is unique because we have to create an instance of the UserData class as it is supposed to be while running so we initialize the class

UserData userData{}

We then use a while loop and a switch case to take all the input, and verify that it is correct and within the specified parameters

```
While (true)
Try {
Display
Cout >> Enter Data Here >> Endl;
Throw
Please Enter correct input!
SetData if no case is thrown
```

4. BalanceAndDeposit.cpp

We now create the backend of the program for the mathematical functions to calculate the output

Using the variables we initialized in step 2 we calculate the corresponding output with a for loop

```
YearEndBalance = initInvest
```

Cool ascii art

```
For (I = 1; I <= years; I++) {
YearEndInterest = yearEndBalance * ((annualInterest / 100))
YearEndBalance = yearEndBalance + yearEndInterest
Output data
```

We then do this a second time just with the addition of monthly deposits but for this because we are using monthly deposits we need to use a second for loop for months to calculate the output properly

```
MonthEndBalance = initInvest
YearEndBalance = initInvest
Const int months = 12
```

Cool ascii art

```
For (I = 1; I <= years; I++) {
YearEndInterest = 0;
For (j= 1; j <= months; j++)
monthEndInterest = 0;
monthEndBalance = monthEndBalance + monthDeposit;
monthEndInterest = (monthEndBalance * ((annualInterest / 100) / months));
yearEndInterest = yearEndInterest + monthEndInterest;
monthEndBalance = monthEndBalance + monthEndInterest;
```

5. Main.cpp

Finally we can create main.cpp

Here we have to create the object that is going to be used within the program

We then call all of our above functions calling the variables for the object