*C++ Requirements*

*You are not allowed to use any global variables. Use of global variables will result in a grade of zero for part 1.*

Your program must have function main, function calculatePresentValue, three read functions, and a display function. Including main this is six functions.

The calculatePresentValue function must have the following signature:

double calculatePresentValue(double futureValue, double interestRate, int numberYears)

Failure to follow the C++ requirements could reduce the points received from passing the tests.

*General overview*

In part 2 you will be creating multiple functions to calculate the present value.

You may be asking what a "present value" is. Suppose you want to deposit a certain amount of money into a savings account and then leave it alone to draw interest for some amount of time, say 12 years. At the end of the 12 years you want to have $15,000 in the account. The present value is the amount of money you would have to deposit today to have $15,000 in 12 years.

The formula used needs the future value (F) and annual interest rate (r) and the number of years (n) the money will sit in the account, unchanged. You will be calculating the present value (P).

P = F / ( (1 + r) ^ n)

In the above expression the value (1 + r) needs to be raised to the nth power. Assume that ^ is the power function and x^2 is x to the 2nd power (x squared)

You are not allowed to use any global variables. Use of global variables will result in a grade of zero for part 2.

*Three read functions*

You must have functions to read in the future value, the annual interest rate, and the number of years. That would be three different functions. Give these functions meaningful names. Note that the order of the values will be future value, annual interest rate, and number of years.

In all cases you need to return any valid value back to the calling function.

For all three functions you will write out to cout as prompt for an input value. You will read in that value from cin. If the value is invalid (zero or negative) you need to display an error message and reread the value (with another prompt). You need to do this in a loop and continue looping until a valid value has been entered. Only the number of years can be an int value. The rest should be of type double.

Here are the prompts for the three values you need to read in:

Enter future value

Enter annual interest rate

Enter number of years

Note that the interest rate will be a number such as 10 or 12.5. These are to be read in as percentages (10% and 12.5%). You will need to divide these values by 100 to convert them into the values needed in the function (.1 and .125 for the above values). This conversion needs to be done before you call the calculatePresentValue function (see below). If you do the conversion in the calculatePresentValue function your program will fail the unit tests, so do the conversion *before* you call the calculate function.

Here are the error messages you need to display if the values are negative:

The future value must be greater than zero

The annual interest rate must be greater than zero

The number of years must be greater than zero

You will also need a function called calculatePresentValue with the following signature:

double calculatePresentValue(double futureValue, double interestRate, int numberYears)

The calculatePresentValue needs to calculate the present value and return that back to the calling function. The formula for this is above. Note that the annual interest would be .08 for 8%.

*The display function*

You also need a function that displays the present value, future value, interest rate, and number of years. The function needs to display the interest rate as %, so .05 would display as 5.0%. Give your display function a meaningful name. You will be passing a number of values to this function.

Here is the sample output for a present value of $69,046.56, a future value of $100,000, an interest rate of 2.5% and a number of years of 15,

Present value: $69046.56

Future value: $100000.00

Annual interest rate: 2.5%

Years: 15

Note that the present value and future value have two digits of precision to the right of the decimal point but the interest rate only has one digit to the right of the decimal point.

The main function will be the driver for your program.

Your program will only be processing one set of valid values for future value, annual interest rate and number of years.

Get the values for these by calling the read functions you created above.

Once you have the values you need to call your calculatePresentValue. Using the result from the calculatePresentValue and the input values you read in with your read functions you need to call your display function (written above) to display the values.

*The main function*

The main function will be the driver for your program. All of the non-main functions are called from main.

For the following sample run assume the input is as follows:

1000000.0

5.0

25

Your program should output the following:

Enter future value

Enter annual interest rate

Enter number of years

Present value: $295302.77

Future value: $1000000.00

Annual interest rate: 5.0%

Years: 25

Here is an example with some invalid data

Input values:

-100

0

1000000.0

5.0

25

Output:

Enter future value

The future value must be greater than zero

Enter future value

The future value must be greater than zero

Enter future value

Enter annual interest rate

Enter number of years

Present value: $295302.77

Future value: $1000000.00

Annual interest rate: 5.0%

Years: 25

Failure to follow the requirements for lab lessons can result in deductions to your points, even if you pass the validation tests. Logic errors, where you are not actually implementing the correct behavior, can result in reductions even if the test cases happen to return valid answers. This will be true for this and all future lab lessons.

*Expected output*

There are eight tests. The first test will run your program with input and check your output to make sure it matches what is expected. The next three tests are unit tests. The unit tests will directly call the calculatePresentValue function. The compilation of the unit test could fail if your calculatePresentValue function does not have the required signature. The final four tests will run your program with various input values and make sure you are calculating the correct answers.

You will get yellow highlighted text when you run the tests if your output is not what is expected. This can be because you are not getting the correct result. It could also be because your formatting does not match what is required. The checking that zyBooks does is very exacting and you must match it exactly. More information about what the yellow highlighting means can be found in course "How to use zyBooks" - especially section "1.4 zyLab basics".

Finally, do not include a system("pause"); statement in your program. This will cause your verification steps to fail.

*Note:* that the system("pause"); command runs the pause command on the computer where the program is running. The pause command is a Windows command. Your program will be run on a server in the cloud. The cloud server may be running a different operating system (such as Linux).

*Error message "Could not find main function"*

Now that we are using functions some of the tests are unit tests. In the unit tests the zyBooks environment will call one or more of your functions directly.

To do this it has to find your main function.

Right now zyBooks has a problem with this when your int main() statement has a comment on it.

For example:

If your main looks as follows:

int main() // main function

You will get an error message:

Could not find main function

You need to change your code to:

// main function

int main()

If you do not make this change you will continue to fail the unit tests.