Programming with Python Assignment 3

Write three programs, one for each of the problems below. Programs should be named like LastnameFirstnameAsn3x.py, where x is either a, b, or c. The first lines of your program should include a line like: # Firstname Lastname Assignment3x # Description of the program.

Part a. Waves

Create a simple program for beach-goers who just want to see some waves. The program should:

1. Create a variable equal to the string 'v';

2. Ask how many 'waves' the user would like printed;

3. Using a 'for' loop, concatenate as many v's as necessary to the variable created in step 1;

4. After the loop has completed, print the variable from step 1.

Your program should look similar to the example shown below.

How many waves would you like to see? *20*

vvvvvvvvvvvvvvvvvvvv

Part b. Compound Interest Schedule

This problem is related to assignment 1B. In that assignment, you displayed a message with the ending balance of a given investment. In this assignment, your program will print a table of: year #, starting balance, interest, and ending balance, for an investment earning compounded interest. Using a 'for' loop to print this table. The equation for the i^{th} year's ending balance is based on assignment 1B's equation, and is given by:

$$\mathrm{E_i} = \mathrm{B_i}\left(1 + \frac{r}{f}\right)^{f}$$

Where E_i is year i's ending balance,

B_i is year i's beginning balance

r is the annual rate of interest (in decimal, e.g., 5% is .05)

f is the number of times the interest rate compounds (times per year)

The interest earned for a given year is $E_i - B_i$

As with assignment 1B, the program should prompt the user for these 4 inputs: beginning balance (investment), term, interest rate, and compounding. Though term is not in the above equation, it is used to determine how many rows the table may have.

Early withdrawal: In addition to the 4 inputs, the program should also ask if the investment will be withdrawn early, and if so, at the end of which year. In this case, the program should print the table up to that early withdrawal year and calculate and display a 5% early withdrawal penalty on that year's ending

balance. A message with the withdrawal penalty and the adjusted final balance (deducting the penalty) should be displayed after the table. The application example session on next page shows how the program should work.

Your program should look similar to the example shown below.

Amount of starting investment ($): *1000*

Term of investment (# of years): *5*

Interest rate (%): *2*

Compounding frequency (times/year): *12*

Enter y if early termination: *n*

year	start balance	interest	end balance
1	1000.00	20.18	1020.18
2	1020.18	20.59	1040.78
3	1040.78	21.01	1061.78
4	1061.78	21.43	1083.21
5	1083.21	21.86	1105.08

OR

Amount of starting investment ($): *1000*

Term of investment (# of years): *5*

Interest rate (%): *2*

Compounding frequency (times/year): *12*

Enter y if early termination: *y*

Enter early termination year #: *4*

year	start balance	interest	end balance
1	1000.00	20.18	1020.18
2	1020.18	20.59	1040.78
3	1040.78	21.01	1061.78
4	1061.78	21.43	1083.21

Early termination penalty = 54.16

Adjusted final balance (after early withdrawal penalty) = 1029.05

Part c.

Create a password verification process. The program should:

1. Select a password (to be guessed by the user) and save it to a variable (should not be a password you actually use);

2. Ask the user to enter a password;

3. If the password is not correct, allow the user 2 more password entry attempts:

    a. Inform the user that the attempt is incorrect and ask the user to try again;

    b. If the new attempt is correct or the user has tried 3 times total, the loop should end;

4. End the program by informing the user if access is granted or not.

To solve this assignment, use a while loop.

Your program should look similar to the example shown below.

Please enter the password: *try1*

That is incorrect, please try again: *pass1*

Access granted

OR

Please enter the password: *try1*

That is incorrect, please try again: *trytwo*

That is incorrect, please try again: *tryagain*

Access denied