Module 2: Compound Interest

JHU EP 605.206 - Introduction to Programming Using Python

Introduction

In this assignment you will create two small programs that take inputs and compute the compound interest on a mortgage loan. Compound interest grows exponentially, not quadratically or geometrically like we're used to thinking. You will also perform a small amount of algebraic manipulation to derive the interest rate of a loan and explore how Python deals with different data types.

Skills: string input/output, data types, operators, and expressions.

Compound Interest Calculator

Imagine you just took out a loan to purchase a newly built home. The total amount you will pay for the house, including interest, can be calculated using the following formula:

$$A = P(1 + \frac{r}{n})^{nt}$$

That is, the total amount you will pay over the life of the mortgage (A) is equal to the principal amount (P), times 1 plus the interest rate (r) divided by the number of times interest is applied per year (n), raised to the power of n times the number of total payments over the life of the loan in years (t).

For example, if you bought a home that cost $500,000 with a 3.0% interest rate (where interest is applied once per month) for a term of 30 years, you would pay: A = 500000 * (1 + 0.03/12)^(12*30) = **$1,228,421.11**.

Programming Assignment

Part 1

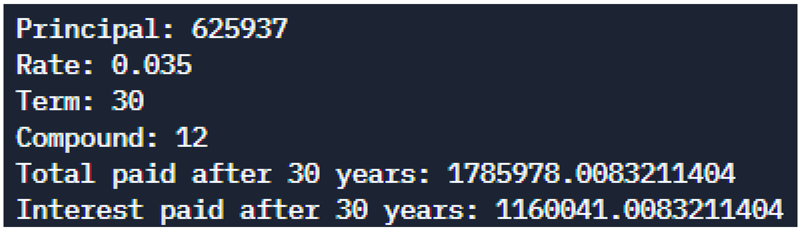
For the first part of the assignment you will write a program named ***interest.py*** to use the formula above to compute the total payment over the life of a loan as well as the amount of interest paid. Your program should prompt a user for the following values:

1. The total principal amount (P, an integer) = 625937
2. The interest rate (r, a float) = 0.035
3. The term of the loan in years (t, an integer) = 30
4. The number of interest payments per year (n, an integer) = 12

Using the operators we learned about this week and the formula for compound interest above, compute the total amount of money that will be paid over the full term of the loan, including interest. Please provide your results in the following format:

> “Total paid after <t> years: <amount>“, where <t> is the term of the loan and <amount> is the total loan cost you calculated in the previous step.
>
> “Interest paid after <t> years: <amount>”, where <t> is the term of the loan and <amount> is the amount of interest that will be paid on the loan.
>
> NOTE: You do not have to perform any rounding on your results.

Here is a sample output for reference (your input prompts can say whatever you’d like):



```
Principal: 625937
Rate: 0.035
Term: 30
Compound: 12
Total paid after 30 years: 1785978.0083211404
Interest paid after 30 years: 1160041.0083211404
```

Part 2

For the second part of the assignment you will write a program named ***rate.py*** to use a modified version of the formula above to calculate the interest rate given the total amount paid, the principal, the term, and the number of interest payments.

First, you will need to perform some algebraic manipulation on the formula used in the previous section. Here it is once more for reference:

$$A = P(1 + \frac{r}{n})^{nt}$$

Using algebra, form a solution that specifies r in terms of A, P, n, and t. That is, re-arrange the equation to solve for r. Once you have your solution, prompt the user for the following:

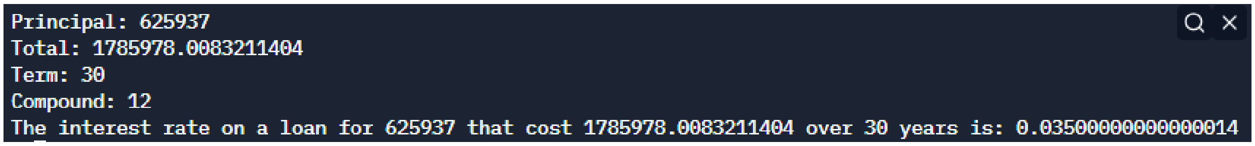
1. The total principal amount (P, an integer) = 625937
2. The total amount paid on the loan including both interest and principal (A, a float) = your result from interest.py
3. The term of the loan in years (t, an integer) = 30
4. The number of interest payments per year (n, an integer) = 12

Using the operators we learned about this week and the formula you derived, compute the interest rate r (a float) given P, A, n, and t. Please provide your results in the following format:

> "The interest rate on a loan for <P> that cost <A> over <t> years is: <rate>", where <P> is the principal amount, <A> is the total amount paid including both interest and principal (the result of interest.py), and <t> is the term of the loan in years.

NOTE: As above, you do not have to perform any rounding on your results.

Here is a sample output for reference (your input prompts can say whatever you'd like):



```
Principal: 625937
Total: 1785978.0083211404
Term: 30
Compound: 12
The interest rate on a loan for 625937 that cost 1785978.0083211404 over 30 years is: 0.035000000000000014
```

<u>Please answer the following question</u>: In theory, your result from rate.py should match the input rate for Part 1. Do they match, yes/no? If not, what is causing the discrepancy?

Deliverables

readme.txt

So-called "read me" files are a common way for developers to leave high-level notes about their applications. Here's an example of a README file for the Apache Spark project. They usually contain details about required software versions, installation instructions, contact information, etc. For our purposes, your readme.txt file will be a way for you to describe the approach you took to complete the assignment so that, in the event you may not quite get your solution working correctly, we can still award credit based on what you were trying to do. Think of it as the verbalization of what your code does (or is supposed to do). Your readme.txt file should contain the following:

1. **Name**: Your name and JHED ID
2. **Module Info**: The Module name/number along with the title of the assignment and its due date
3. **Approach**: a detailed description of the approach you implemented to solving the assignment. Be as specific as possible. If you are sorting a list of 2D points in a plane, describe the class you used to represent a point, the data structures you used to store them, and the algorithm you used to sort them, for example. The more descriptive you are, the more credit we can award in the event your solution doesn't fully work.
4. **Known Bugs**: describe the areas, if any, where your code has any known bugs. If you're asked to write a function to do a computation but you know your function returns an incorrect result, this should be noted here. Please also state how you would go about fixing the bug. If your code produces results correctly you do not have to include this section.

Please submit your ***interest.py*** and ***rate.py*** source code files along with a PDF file containing screenshots of your 2 outputs and your answer to the question in Part 2 in a PDF called JHEDID_mod2.pdf (ex: jkovba1_mod2.pdf). Please do not ZIP your files together.

Recap:

1. readme.txt
2. interest.py
3. rate.py
4. 2 screenshots of your outputs from each .py file
5. Answer to the question from Part 2

Please let us know if you have any questions via Teams or email!