CS 115, Autumn 2021
Programming Project #1: Loan Calculator (10 points)
Due Tuesday, September 28th 2021, 11:30 PM

This assignment tests your understanding of `print` statements, escape sequences and variable. Write a Python program in files named `about_me.py` and `loan_calculator.py`. (Use exactly these file names, including identical capitalization.)

Part 1: `about_me.py`

For this part of the assignment, write a program that prints out information about you in a file called `about_me.py`. You will receive full credit if your program meets the following requirements and the requirements listed below in the style section:

1. At least 3 separate paragraphs that each contain at least 2 sentences. Leave blank lines between paragraphs. You may indent your paragraphs if you wish.
2. Include at least 10 escape sequences to print at least 3 different characters.
3. Include your name, why you are taking CS 115, what you hope to do with computer science and 2 random facts about yourself. You are welcome to include any other information or text drawings you like as well as long as they are school appropriate.
4. All prints should be in a `main` function and the `main` function should be called at the end of the file.

Part 2: `loan_calculator.py`

For this part of the assignment, write a program that prompts the user for loan information and then outputs their monthly payment. Write your code in a file called `loan_calculator.py`.

```
/\'/\"/\ CS 115 Interest Calculator /\"/\'/\

Welcome to the CS 115 interest calculator!
Enter your loan information to find out what
your monthly loan payment will be.
Enter all values as whole numbers.
Do not include ., ', ", $ or % characters.

How much money is your loan for? 440000
What is the interest rate on your loan (enter 3% as 3)? 3.875
How many payments do you make per year? 12
How many years is your loan for? 30

You will have a loan payment of $2069.0431690119103 each month.
Overall you will pay $744855.5408442877.
This is $304855.5408442877 more than the amount you borrowed.
```

The box to the left contains an example of what your program should output. User input is shown as bold and underlined. You do not need to make your text bold or underlined, this is just to make it easier to see which pieces are user input.

Your program must **exactly** reproduce the output at left when the user types in the same input. This includes identical wording, spelling, spacing, punctuation, and capitalization. We will be very strict about this so make sure you use the output comparison tool on the course website to check you have the correct output.

Use the following equation to calculate the loan payment amount:

$$payment = principle * \frac{interestPerPayment * (1 + interestPerPayment)^{totalPayments}}{(1 + interestPerPayment)^{totalPayments} - 1}$$

`payment`: the amount the user will pay per month
`principle`: the amount the loan is for
`interestPerPayment`: the interest percentage on the loan converted to a percentage (divided by 100) and divided by the number of payments you make per year
`totalPayments`: the total number of payments you will make over the entire time you have the loan

In the example on page 1 these are the values used for each part of the equation:

- `principle = 440000`

- `interestPerPayment = 3.857 / 100 / 12`
- `totalPayments: = 12 * 30`

Development Strategy:

It can be a bit overwhelming to figure out how to start writing a program. Therefore, we have provided some steps below to help you get started. While you do not have to follow this list of steps, We think you will be more successful if you do.

1. Copy the contents of one of the expected output files into a blank Python file. Put each line (including the user input and computed values) into a `print` statement. Escape any characters necessary so that your program outputs the exact text of one expected output.
2. Change the `print`s you have used to output the four lines that prompt the user for input to `input` statements. Make sure you store the result of these into variables, converting them to number types if you will want to use them in equations later. Test that you have done this correctly by printing out each of the variables. You should see the same numbers appearing that you typed in.
3. Compute the loan payment. Use a variable to store the result of each operator. Use these variables in future steps of computing the result of the equation so that you do not have to repeat the computation.
4. Compute the overall amount the user will pay by multiplying the monthly payment by the number of years and the number of payments per year. Note that this will be greater than the principle as the user will be charged interest on the loan.
5. Compute the extra the user will pay by having the loan by subtracting the principle from the number you computed in step 4 of this list.

It can sometimes be hard to figure out what is going wrong if you get an incorrect result from a complex equation. Therefore, I have written out below what value you should have at every step in your computation for the example on page 1. Make sure your variables store the same numbers.

Example:

You will find a step-by-step guide to computing the answer to the example on page 1 below. Evaluate the expression in this order, evaluating the yellow highlighted operands and operators at each step. Use variables to store the result of each computation. Note that your last couple digits of long decimals may not match exactly because of imprecision – this is fine.

$$payment = 440000 * \frac{(3.857 / 100 / 12) * (1 + (3.857 / 100 / 12))^{(12*30)}}{(1 + (3.857 / 100 / 12))^{(12*30)} - 1}$$

$$payment = 440000 * \frac{(3.857 / 100 / 12) * (1 + (3.857 / 100 / 12))^{360}}{(1 + (3.857 / 100 / 12))^{360} - 1}$$

$$payment = 440000 * \frac{0.00321416666 * (1 + 0.00321416666)^{360}}{(1 + 0.00321416666)^{360} - 1}$$

$$payment = 440000 * \frac{0.00321416666 * (1.00321416666)^{360}}{1.00321416666^{360} - 1}$$

$$payment = 440000 * \frac{0.00321416666 * 3.17479929609}{3.17479929609 - 1}$$

$$payment = 440000 * \frac{0.01020433404}{2.17479929609}$$

$$payment = 440000 * 0.0046920808$$

$$payment = 2069.0431690119103$$

Style Guidelines:

All of your code must be in a file that is runnable. You should **not** type each of your statements into the interpreter and run them one at a time. All of your statements in a file should be inside a `main` function and that function should be called at the bottom of the file.

Each line of output should be produced by a separate `print` or `input` statement. Do not use one `print` to output several lines of output. Likewise, do not compute the monthly payment in one big expression. If you try to do this your expression will be very long, complex and errors will be harder to find. Instead, store each piece of the equation in its own **variable**. Give these variables names that describe what they store so your code is easier to understand. Names like `a, b, c, variable1, variable2, variable3, pmt, prple, ipp` and `tp` are not good variable names. The first six do not help you figure out what they should store. The last three are abbreviations one could use for some of the pieces of the equation described above. However, they are still hard to understand as they are so short. It is considered better Python style to use full words in variable names. If your names include multiple words separate them with an underscore. All names should be all lowercase.

We do not like to do extra work in computer science. Therefore, if we compute a result we do not want to have to compute it again. If you use a value twice in a large equation, store its result in a variable and use that same variable multiple times. Do not create two separate variables.

Include a comment at the beginning of your program with some basic information and a description of the program in your own words. For example:

```
# Suzy Student, CS 115, Autumn 2049
# Programming Project #1, 06/07/49
#
# This program's behavior is ...
```

For this assignment, you should limit yourself to the Python features covered in the first five lectures. Though we will cover more material while you work on this assignment, please do not use any of it in this program, such as `if/else` statements.

Submission and Grading:

Turn in your Python source code file electronically from the Project page on the course web site.

Part of your program's score will come from its "external correctness." External correctness measures whether the output matches exactly what is expected. We are very picky about the output matching exactly and expect every character and space to match. Use the **output comparison tool** to help you make sure your output is perfect. Programs that do not run will receive no external correctness points.

The rest of your program's score will come from its "internal correctness." Internal correctness measures whether your source code follows the stylistic guidelines specified in this document. This includes having an adequate comment header and using `print`, `input` and variables well. You should make sure to name your variables with descriptive names in all lowercase and underscores between words. You should also limit the lengths of all lines in your program to **fewer than 80 characters**.