CS10 Python Programming Homework 1
(20 points)

1. You must turn in your program listing and output for each program set. Each program set must have your student name, student ID, and program set number/description. Late homework will not be accepted for whatever reasons you may have.

*********_**for this homework, you are to submit your Program sets to Canvas**_ under Homework 1 link*********

   a. Name your files : HW1_PS1_ lastname_firstinitial.py for Program Set 1 and HW1_PS2_ lastname_firstinitial.py for PS2 and so on. PS means program set. If there are two program sets you will submit two files one for each program set. Example if your name is Joe Smith then the filename will be HW1_PS1_smith_j.py
   b. You must submit your homework by the deadline at the specified upload link on Canvas under homework 1. If the deadline is past, Program Sets will not be graded. Homework submitted via email attachment, comment in Canvas, Canvas message, or by any other method is not accepted and will be given a zero for no submission.
   c. if you do not follow the instructions on file naming provided in this section you will receive a zero for that program set that you did not correctly name the file.
   d. It is your responsibility to check if your homework is properly submitted to Canvas.

2. Please format your output properly, for example all dollar amounts should be printed with 2 decimal places when specified. Make sure that your output values are correct (check the calculations).

3. Use only the 'tools' in the topics we covered from lesson 1 to lesson 10 only. **No if statements, loops, list, dictionary, classes and any other topics not covered in lesson 1 to 10. Using any 'tools' or topics not taught in lesson 1 to 10 will result in a zero for that program set.**

4. Each student is expected to do their own work. **IF IDENTICAL PROGRAMS ARE SUBMITTED, EACH IDENTICAL PROGRAM WILL RECEIVE A SCORE OF ZERO.**

Grading:

Each program set must run correctly both syntactically, logically, and display the correct output as specified. **If the program set does not run, a zero will be given**. If the program executes properly with proper syntax, logic, and displays the correct output with proper formatting as specified in the question, you will receive the full points for that question. Then points will be **deducted for not having proper**:

   a. Comments (1 pt deducted for not having comments as listed below)
      - Your name, description at the beginning of each program set.
      - Short description of what each section of your codes does.

accounts payable
acct_pay

   b. Consistency/Readability (2 pts for each infraction)
      - Spacing(separate each section of codes with a blank line)
      - Indentation only when needed
      - Proper naming of variables **no a, b, c** – use descriptive and mnemonics)

   c. Required elements (2 pts for each infraction)
      - proper formatting for output when specified
      - all monetary values must be in 2 decimal places

   d. Use only 'tools' in the topics that have been covered in class. For example, in for this homework you are only allowed to use topics covered in lesson 1 to 10. If you use if statements, loops, or list, and tools not taught in lesson 1 to 10 your program set will result in a zero score.

e. Output (you **must provide the specified number of test runs or your program set will receive a zero score**)
   - to be displayed at the end of the program listing(codes) and commented
   - if no output(test runs) is provided from your uploaded file, a zero will be given for that program set.
   - must have the number of test runs as specified in each program set.
   - must use the data for test runs when they are provided for you in the question.

Points will be deducted from items a. to e. above until your Program Set reaches zero points.

Program Set 1 (10 points)

Write a program to compute the monthly payment and total payment. The program will ask the user for the annual interest rate and the number of years and loan amount.

Here are the formulae needed:

monthlyInterestRate = annual interest rate / 1200

monthlyPayment = loanAmount * monthlyInterestRate / (1 - 1 / (1 + monthlyInterestRate) ** (numberOfYears * 12))

totalPayment = monthlyPayment * numberOfYears * 12

Sample program run below, user input in blue. Run program 4 more times with different user input and its output. Check each output with an online amortization calculator to verify your program runs correctly. You will provide your own data for test run 2, 3, 4 , and 5 . Use test run 1 data that was provided for you below. Include test run 1 in your output to make up to 5 test runs for your submission. Output must look exactly as shown below.

Here is a sample test run(blue user input):

>>>
Test Run 1

Enter annual interest rate, (e.g., 7.25) : 4.5
Enter number of years as an integer, (e.g., 5) : 30
Enter loan amount, (e.g., 120000.95) : 350000.50

The monthly payment is 1,773.40
The total payment is 638,424.40

Test Run 2
Run the program again and enter different user value and paste it here

Test Run 3
Run the program again and enter different user value and paste it here

Test Run 4
Run the program again and enter different user value and paste it here

Test Run 5
Run the program again and enter different user value and paste it here

>>>

Program Set 2 (10 Points)

Stock Transaction Program

Last month Kool Doode purchased some stock from Kaplack, Inc.

Write a program that ask the user to input the followings:

1. The stock name
2. Number of shares Kool Doode bought
3. Stock purchase price
4. Stock selling price
5. Broker commission

displays the following paid for the stock.
:

1. The amount of money Kool Doode paid for the stock (number of shares bought * purchase price)
2. The amount of commission Kool Doode paid his broker when he bought the stock. (Amount he paid for stocks * commission in percent)
3. The amount that Kool Doode sold the stock for. (number of shares * selling price)
4. The amount of commission Kool Doode paid his broker when he sold the stock. (Amount he sold shares * commission in percent)
5. Display the amount of money Kool Doode had left when he sold the stock and paid his broker (both times). If this amount is positive, then Kool Doode made a profit. If the amount is negative, then Kool Doode lost money.
   Profit/loss =(amount for sold stocks- commission) - (amount paid to buy stocks + commission)

Two sample program runs provided below, user input in blue. Perform 3 more test runs with different user input and its output. Check each output with calculator to verify your program runs correctly. Output must look exactly as shown below including all proper alignments.

Assume that the user will always enter the Stock Name as a string and not anything else. The output amount for amount paid and amount sold will not exceed 9,999,999.99

10 blank spaces — 13 columns → use format( ) to specify the column width and decimal places.

```
Amount paid for the stock:          $    339,200.00
Commission paid on the purchase:    $     13,568.00
Amount the stock sold for:          $    359,200.00
Commission paid on the sale:        $     14,368.00
Profit (or loss if negative):       $     -7,936.00
```

Here are 2 sample runs(blue user input):

```
##Test Run 1
##Enter Stock name: Kaplack, Inc.
##Enter Number of shares : 10000
##Enter Purchase price : 33.92
##Enter selling price : 35.92
##Enter Commission : 0.04
##
##
##Amount paid for the stock:          $    339,200.00
##Commission paid on the purchase:    $     13,568.00
##Amount the stock sold for:          $    359,200.00
##Commission paid on the sale:        $     14,368.00
##Profit (or loss if negative):       $     -7,936.00
```

```
##Test Run 2
##Enter Stock name: IBM
##Enter Number of shares : 15000
##Enter Purchase price : 50.25
##Enter selling price : 100.20
##Enter Commission : 0.02
##
##
##Amount paid for the stock:          $   753,750.00
##Commission paid on the purchase:    $    15,075.00
##Amount the stock sold for:          $ 1,503,000.00
##Commission paid on the sale:        $    30,060.00
##Profit (or loss if negative):       $   704,115.00
##
##Test Run 3
##Run the program again and enter different user value and paste it here
##
##Test Run 4
##Run the program again and enter different user value and paste it here
##
##Test Run 5
##Run the program again and enter different user value and paste it here
```

You will provide your own data for test run 3, 4 , and 5 . Use the data from sample test runs 1 and 2 that was provided for you above. Include test runs 1 and 2 in your output to make up to 5 test runs for your submission.