CS10 Python Programming Homework 2
20 points (Lesson 1 to Lesson 25)

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 2 link*********

   a. Name your files : HW2_PS1_ lastname_firstinitial.py for Program Set 1 and HW2_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 HW2_PS1_smith_j.py
   b. You must submit your homework by the deadline at the specified upload link on Canvas under homework 2. 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 the question 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 25 (no functions) only. **No functions, lists, dictionary, classes and any other topics not covered in lesson 1 to 25. Using any 'tools' or topics not taught in lesson 1 to 25 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 exactly as specified. **If the program set does not run and does not display the output exactly as specified shown in my sample test 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 program set, 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.

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)

accounts payable
accts_pay

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 25. If you use list and/or tools not taught in lesson 1 to 25 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.

USE ONLY THE 'TOOLS' OR TOPICS YOU HAVE BEEN TAUGHT IN CLASS, IF YOU USE ANYTHING, WE HAVE NOT COVERED YET YOU WILL RECEIVE A ZERO FOR THAT PROGRAM SET . Use 'tools' or topics up to loops and not beyond, no lists, functions. The break and continue command is not allowed. If you use break or continue a zero will be given for that program set.

Program Set 1(5 points)

Lottery program. The program randomly generates a two-digit number, prompts the user to enter a **single** two-digit number, and determines whether the user wins according to the following rules. Write a loop to let the user play as many times as the user wanted. Use a sentinel or flag to quit out of the loop. You must pick the correct loop for this question. Each time the user continues to play, the program will generate a new two-digit random number until the user enters -999 to quit.

1. if the user's input matches the lottery in the exact order, the award is $10,000.

2. if all the digits in the user's input match all the digits in the lottery number, the award is $3,000.

3. if one digit in the user's input matches a digit in the lottery number, the award is $1,000.

Sample Test Run 1(blue user input) Since this is a random number program, if you use the same input as my sample run it will not produce the same results, therefore, use your own input values for your 5 test runs.

```
Enter your lottery pick ( 2 digits) or -999 to quit: 44
Sorry no match

Enter your lottery pick ( 2 digits) or -999 to quit: 23
Match one digit:  You win $1,000

Enter your lottery pick ( 2 digits) or -999 to quit: 68
Match one digit:  You win $1,000

Enter your lottery pick ( 2 digits) or -999 to quit: 12
Match all digits :  You win $3,000

Enter your lottery pick ( 2 digits) or -999 to quit: 45
Exact match:  You win $10,000!

Enter your lottery pick ( 2 digits) or -999 to quit: -999
>>>
```

the random number is 35
user enters 44 - no match

the random number is 38
user enters 23 - match one digit

the random number is 62
user enters 68 - match one digit

the random number was 21
user enters 12 - match all digits
but not exact match

the random number is 45
user enters 45 - exact match

Provide your own test runs 2, 3, 4, 5.

Program Set 2 (15 Points)

This assignment will give you more experience on the use of:

1. integers (int)
2. floats (float)
3. conditionals(if statements)
4. iteration(loops)

The goal of this project is to make a fictitious comparison of the federal income. You will ask the user to input their taxable income. Use the income brackets given below to calculate the new and old income tax. For the sake of simplicity of the project we will only consider individuals and not married users. We will also ignore any tax deductions while calculating income tax—they can significantly alter the tax, but add too much complexity for our programming project.

New income tax brackets (2018 and newer)

Rate	**Income range**
10%	Up to $9,525
12%	$9,526 to $38,700
22%	$38,701 to $82,500
24%	$82,501 to $157,500
32%	$157,501 to $200,000
35%	$200,001 to $500,000
37%	over $500,000

Old income tax brackets (2017 and older)

Rate	**Income range**
10%	Up to $9,325
15%	$9,326 to $37,950
25%	$37,951 to $91,900
28%	$91,901 to $191,650
33%	$191,651 to $416,700
35%	$416,701 to $418,400
39.6%	over $418,400

Assignment Background

Being in the 25% tax bracket doesn't mean you pay 25% on everything you make. The progressive tax system means that people with higher taxable incomes are subject to higher tax rates, and people with lower taxable incomes are subject to lower tax rates.

For example, let's say you're a filer with $32,000 in taxable income. That puts you in the 15% tax bracket in 2017. But do you pay 15% on all $32,000? No. Actually, you pay only 10% on the first $9,325; you pay 15% on the rest. (Look at the tax brackets above to see the breakout.)

If you had $50,000 of taxable income, you'd pay 10% on that first $9,325 and 15% on the chunk of income between $9,326 and $37,950. And then you'd pay 25% on the rest, because some of your $50,000 of taxable income falls into the 25% tax bracket. The total bill would be $8,238.75 — about 16% of your taxable income, even though you're in the 25% bracket.

Project Description

Your program must meet the following specifications:
1. At program start, prompt the user for their income
2. Repeatedly prompt the user for new income until a negative income is entered.
3. Calculate the income tax using the 2017 and 2018 tax bracket tables above.
4. For each income entered:
    a. Calculate the 2017 income tax and store it in a variable.
    b. Next calculate the 2018 income tax and store it in a variable
    c. Print
        i. The income
        ii. The 2017 tax
        iii. The 2018 tax
        iv. The difference between the 2018 and 2017 tax rounded to cents.
        v. The difference between the 2018 and 2017 tax as a percentage of the 2017 tax rounded to cents

Assignment Notes

1. To clarify the project specifications, sample test runs output is appended to the end of this document.
2. Use a while loop that keeps looping as long as the income is greater than or equal to zero. Prompt for income before the loop
4. There will be no error checking in this assignment. If a float or a string is entered at the prompt, the program will crash. Therefore, you will only enter integers for this program.
5. All income input must be integers.

Test Runs

Sample Test Run 1

```
Enter income as an integer with no commas: 8000
Income: 8000
2017 tax: 800.00
2018 tax: 800.00
Difference: 0.00
Difference (percent): 0.00

Enter income as an integer with no commas: 15000
Income: 15000
2017 tax: 1783.75
2018 tax: 1609.50
Difference: -174.25
Difference (percent): 9.77

Enter income as an integer with no commas: 40000
Income: 40000
2017 tax: 5738.75
2018 tax: 4739.50
Difference: -999.25
Difference (percent): 17.41
```

```
Enter income as an integer with no commas: 100000
Income: 100000
2017 tax: 20981.75
2018 tax: 18289.50
Difference: -2692.25
Difference (percent): 12.83

Enter income as an integer with no commas: 200000
Income: 200000
2017 tax: 49399.25
2018 tax: 45689.50
Difference: -3709.75
Difference (percent): 7.51

Enter income as an integer with no commas: 500000
Income: 500000
2017 tax: 153818.85
2018 tax: 150689.50
Difference: -3129.35
Difference (percent): 2.03

Enter income as an integer with no commas: 1000000
Income: 1000000
2017 tax: 351818.85
2018 tax: 335689.50
Difference: -16129.35
Difference (percent): 4.58

Enter income as an integer with no commas: 10000000
Income: 10000000
2017 tax: 3915818.85
2018 tax: 3665689.50
Difference: -250129.35
Difference (percent): 6.39

Enter income as an integer with no commas: -1
```

Test Run 2 (This test run 2 is to test when user enters -1 as income and the program should quit, see specifications)

```
Enter income as an integer with no commas: -1
```

Run test runs 3,4,5 using different income values(each test run 3 to 5 must have at least 3 incomes entered before quitting the run with -1).