CP 104 – Introduction to Programming

Assignment 02
Due: Check Moodle

General Assignment Notes. Please read carefully.

Your assignments must follow the following requirements:

- Name your project with:
  - your login,
  - an underscore,
  - 'a' (for 'assignment'),
  - the assignment number: *login_a#.*

- For example, if a student with login *barn4520* submits Assignment 1, the name should be: *barn4520_a1* or *barn4520_a01*.
-
- In your program, use the variable naming style given in Coding and Documentation Style https://peps.python.org/pep-0008/

- Zip the entire project. Give your .zip file the same name as your project when exporting your project, e.g. barn4520_a1.zip .

- Unless otherwise indicated by the question you may only use the built-in functions and special forms introduced in the lecture slides

- The solutions you submit must be entirely your own work. Do not look up either full or partial solutions on the Internet or in printed sources.

- Where a sample run of the program is provided, the underlined values are entered by the user.

General Marking Expectations

- Although the marking scheme is tailored for each question, you can use the following as an indication of what we are looking for when marking your assignments.
  - General:
    - project and zip files named correctly. Using the wrong names is an automatic zero.
    - uses the variable naming style given in Coding and Documentation Style Standards : i.e. lower case variable names, underscores between words
  - main:
    - identification template included and filled in correctly
    - inputs as required for program
    - outputs as required for program

CP 104 – Introduction to Programming

Task 01: Write and test a python program **t01.py** that shows your understanding of how single, double, and triple quotes can be used by using four (4) `print` functions to:

1. display the text: I asked him to study the book ‘Starting out with python’ from “Tony Gaddis”.

2. display the text: I’m going to get some bread for sandwiches.

3. display the text: "You have enemies? Good. That means you've stood up for something, sometime in your life." Winston Churchill

4. display the text: Exact naming of files, functions and variable names is critical for all the assignments. Auto-grading will not work unless I have precisely followed the rules and names given in the assignment document.

You may use only one print function per piece of text. Use no other print functions in your program - no titles, no blank lines, nothing. **t01.py** will have exactly 4 lines of code. Your solution *must* include at least one use of single quotes, one use of double quotes, and one use of triple quotes. You may not use any escape characters in your print functions i.e you may not use '\n' or quote escaping: \" or \'.

Task 02: Write and test a program **t02.py** that converts gallons to liters and displays it . The value for gallons is entered by the user. *Hint: 1 gallon = 3.78541 liter*

Sample Run of the program:

```
Enter gallons: 20
20 gallons is equivalent to 75.7082 liters.
```

- Define constants when appropriate.

Task 03: Write and test a program **t03.py** that calculates and displays the double and triple of a number entered by the user.

Sample Run of the program:

```
Enter number: 5
The double and triple of the number is:   10 and 15
```

Task 04: Write and test a program **t04.py** that displays a table of a number entered by the user.

Sample run of the program:

```
Enter a number to display its multiplication table: 5
5    x    1    =    5
5    x    2    =    10
5    x    3    =    15
5    x    4    =    20
5    x    5    =    25
```

```
5	x	6	=	30
5	x	7	=	35
5	x	8	=	40
5	x	9	=	45
5	x	10	=	50
```

• Tab ("\t") is used for spacing in the above table.

Task 05:
A vineyard owner is planting several new rows of grapevines, and needs to know how many grapevines to plant in each row. She has determined that after measuring the length of a future row, she can use the following formula to calculate the number of vines that will fit in the row, along with the trellis end-post assemblies that will need to be constructed at each end of the row:

$$V = (R - 2E) / S$$

The terms in the formula are:
V is the number of grapevines that will fit in the row.
R is the length of the row, in feet.
E is the amount of space, in feet, used by an end-post assembly. *S* is the space between vines, in feet.

Write a program **t05.py** that makes the calculation for the vineyard owner. The program should ask the user to input the following:

- The length of the row, in feet
- The amount of space used by an end-post assembly, in feet
- The amount of space between the vines, in feet

Once the input data has been entered, the program should calculate and display the num-ber of grapevines that will fit in the row.

Sample Run of the program:

```
Enter the length of the row, in feet: 5
Enter the amount of space, in feet, used by an end-post assembly: 1
Enter the distance, in feet, between each vine: 1
You have enough space for 3.0 vines.
```

Note:
Exact naming of files, functions and variable names is critical.
You cannot use any concept not yet taught in the class like if-else statements or loops.