CS1026: Assignment 1
(Coffee or Tea, I see)

Due: Wednesday, October 3th, 2018 at 9:00pm
Weight: 5%

Learning Outcomes:
By completing this assignment, you will gain skills relating to

- basic Python programming constructs,
- expressions and decisions,
- getting input from user,
- validating input,
- algorithm development and testing,
- following program specifications.

Task:
In this assignment, you will write a **complete** program in Python that computes the cost of a specialty coffee or tea beverage. Your program is expected to prompt the user for input and validate it before computing the results. Your program should make use of expressions, decisions and basic input/output in Python.

Functional Specifications:

1. The program will prompt the user for various pieces of information about the desired beverage. The information required is described below; some of the information is dependent on the type of beverage ordered. Once all the information has been entered, the program will compute and display the amount of money charged for that customer's beverage.

The program will prompt the user to enter the following (in the specified order):

a. The customer's name (a string);
b. The type of beverage (coffee or tea) (a string);
c. The size of the beverage (small, medium, large) (a string).
d. Any added flavorings:
    - For coffee this could be none, vanilla or chocolate.
    - For tea this could be none, lemon or mint.

It will then process that customer's information and display the results.

2. The input from the user should be as follows:

a. The customer's name – A string consisting of **only upper and lower case letters**; no spaces (***you may assume that it only contains letters of the alphabet***).
b. Type of beverage (coffee or tea):

- For coffee: “C”, “c”, “coffee” or **ANY** combination of upper and lower case letters that correctly spells “coffee”, i.e., “Coffee”, “COFFEE”, “coffEE” are acceptable; “Café” is **NOT**.
- For tea: “T”, “t”, “tea” or **ANY** combination of upper and lower case letters that correctly spells “tea”, i.e., “Tea”, “tEA” are acceptable; “Te” and “TEE” are **NOT**.

c. Size of the beverage: ***small*** (“small” , “S”, “s”), ***medium*** (“medium”, “M”, “m”), ***large*** (“large”, “L”, “l”) or **ANY** combination of upper and lower case letters that correctly spells “small”, “medium” or “large”.

d. Flavoring:
- For none: a RETURN or empty line or the word “none” or **ANY** combination of upper and lower case letters that correctly spells “none”.
- For vanilla: “V”, “v”, “vanilla” or **ANY** combination of upper and lower case letters that correctly spells “vanilla”.
- For chocolate: “C”, “c”, “chocolate” or **ANY** combination of upper and lower case letters that correctly spells “chocolate”.
- For lemon: “L”, “l”, “lemon” or **ANY** combination of upper and lower case letters that correctly spells “lemon”.
- For mint: “M”, “m”, “mint” or **ANY** combination of upper and lower case letters that correctly spells “mint”.

3. The program will compute the amount of money that the customer owes for the beverage; the cost depends on the size of the beverage and any additional flavorings (NOTE: the customer gets a choice of only one flavoring!). The costs are as follows:

a. Size of beverage:
   i. Small: $1.50
   ii. Medium: $2.50
   iii. Large: $3.25

b. Coffee flavorings:
   i. Vanilla: $0.25
   ii. Chocolate: $0.75

c. Tea flavorings:
   i. Lemon: $0.25
   ii. Mint: $0.50

4. The program should compute the total cost of the beverage with additional taxes; taxes are 13%. The final cost, with tax included, should be rounded to the nearest cent. For example, a small coffee with no flavoring would cost the customer a total of: $1.67 (1.50 * 1.11 = 1.665 which rounded to the nearest cent is 1.67 (let’s assume that we still have pennies!). (Note: make sure that the amount of money billed is displayed with a dollar sign and two decimal points).

5. For each customer, the program should display a single line of output describing the beverage and the cost, e.g.:

For Kemi, a small coffee, no flavoring, cost: $1.67.

All output should be appropriately labeled and formatted. The amount of money billed should be displayed with a dollar sign and will be rounded to two fractional digits (for example, $125.99 or $43.86).

6. The program should also detect and report invalid input. ***When an invalid input is detected, the program will display an error message indicating the error, e.g., an invalid size specified. After displaying this information the program should end without prompting for any additional information.***

   Examples of invalid input include entering a size or type of beverage that does not exist. Another example is entering a flavoring that does not exist for that particular beverage. For instance, if a user is getting coffee and enters MINT as the flavor of choice, your program should recognize this as an invalid input.

Non-functional Specifications:

1. Include brief comments in your code identifying yourself, describing the program, and describing key portions of the code.
2. Assignments are to be done individually and **must be your own work**. Software may be used to detect cheating.
3. Use Python coding conventions and good programming techniques, for example:
   i. Meaningful variable names
   ii. Conventions for naming variables and constants
   iii. Use of constants where appropriate
   iv. Readability: indentation, white space, consistency

The name of the file you submit should be your UWO userid_Assign1.py. ***Make sure you attach your python file to your assignment; DO NOT put the code inline in the textbox***.

Make sure that you develop your code with Python 3.6 as the interpreter. TAs will not endeavor to fix code that uses earlier versions of Python.

What You Will Be Marked On:

1. Functional specifications:
   - Does the program behave according to specifications?
   - Does the program handle invalid input?
   - Is the output according to specifications?
2. Non-functional specifications: as described above
3. Assignment submission: via OWL assignment submission