CS1026: Assignment 1
Home Power Consumption

Due: October 6th, 2021, 9pm
Weight: 5%

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

- Basic Python programming constructs
- Expressions, decisions, loops
- Getting input from users
- Algorithm development and testing; designing test cases
- Following program specifications.

Background:
Ontario has several different ways of computing a home owner's cost for electricity. One of these methods is called Time of Use (TOU). This method divides the day into periods: Off Peak (7pm-7am), Peak (7am-11am and 5pm-7pm), Mid-Peak (11am-5pm). The electricity usage, in kilowatt hours (kwh) is measured during each period and charged different rates. The rates are:

- $0.085 per kwh during Off Peak;
- $0.176 per kwh during Peak;
- $0.119 per kwh during Mid Peak.

Tasks:

In this assignment, you will write a **complete** program in Python that computes the cost of electricity for individual residences. Your program should account for several discounts based on usage and whether the residence belongs to a senior or not. The discounts to be considered are the following:

- *Total Usage Discount*: If the total kwh used are less than 400, then there is a 4% discount on the price; this is before taxes.
- *On Peak Discount*: If the kwh used during the On Peak period are less than 150, then there is a discount of 5% on the cost of the electricity used during the On Peak period. This is NOT applicable if the *Total Usage Discount* is applied.
- *Senior Discount*: If the home owner is a senior, then there is an 11% discount on the total cost after any of the other discounts have been applied, but before taxes.

After the total cost is determined, there is a tax of 13%.

Your program is expected to prompt the user for input and compute the results. Your program will make use of expressions, decisions, and input/output in Python. You should name your program **Assign1.py**. **Your program should strictly adhere to the Functional Specifications (below) and Non-Functional Specifications**.

Functional Specifications:

1. Your program will compute the cost of electricity for individual residences. It will prompt the user for input about electricity consumption, print the cost and continue prompting until a value of 0 is entered.

   The program will prompt the user for the information on the power consumption of a residence and whether the owner is a senor or not. Once all the information has been entered, the program will compute and display the cost of electricity (including and discounts and tax).

   The program will prompt for the power consumption for the Off Peak period, Peak period, Mid Peak period and whether the owner is a senior or not. The program should prompt for these values in the order described. An example of the program's input is:

   ```
   Enter kwh during Off Peak period: 120.0
   Enter kwh during On Peak period: 160.7
   Enter kwh during Mid Peak period: 312.7
   Is owner senior? (y,n):n
   ```

   If the amount entered for the Off Peak period is 0 (0.0), then the program will halt. If a 0 is entered, your program should **NOT PROMPT for any other inputs**.

2. The input indicating whether the owner is a senior or not should be one of "Y", "y", "N", or "n".

3. The output consists of a single line with the cost for the electricity for that residence. The output should be printed using Python formatting operators and should print to 2 decimal places using the formatting operators; **the output MUST be identified with the phrase "Electricity cost:"** as in the example below:

   ```
   Electricity cost: $85.53
   ```

4. Assumptions. You may assume the following when writing your program:
   - **a.** The input order is as defined above, that is: Off Peak kwh, On Peak kwh, Mid Peak kwh, senior or not.
   - **b.** The kwh can be floating numbers.
   - **c.** You may assume that the values entered are valid – that is the kwhs are floating numbers greater than 0 and that the specification of the owner is one of "Y", "y", "N", or "n". You do not need to check for valid input.

5. Finally, an automated testing program will run a number of test cases against your program. Some examples of output and test cases are presented below; **these are NOT**

comprehensive - you should create your own test cases and thoroughly test your program.

Example executions and output:

# Example 1

```
Enter kwh during Off Peak period: 120.0
Enter kwh during On Peak period: 160.7
Enter kwh during Mid Peak period: 312.7
Is owner senior? (Y,y,N,n):n
Electricity cost: $85.53

Enter kwh during Off Peak period: 120.0
Enter kwh during On Peak period: 160.7
Enter kwh during Mid Peak period: 312.7
Is owner senior? (Y,y,N,n):y
Electricity cost: $76.13

Enter kwh during Off Peak period: 0

Process finished with exit code 0
```

# Example 2

```
Enter kwh during Off Peak period: 156.7
Enter kwh during On Peak period: 98.6
Enter kwh during Mid Peak period: 255.5
Is owner senior? (Y,y,N,n):n
Electricity cost: $68.04

Enter kwh during Off Peak period: 112.3
Enter kwh during On Peak period: 198.4
Enter kwh during Mid Peak period: 406.3
Is owner senior? (Y,y,N,n):n
Electricity cost: $104.88

Enter kwh during Off Peak period: 132.4
Enter kwh during On Peak period: 151.5
Enter kwh during Mid Peak period: 345.5
Is owner senior? (Y,y,N,n):y
Electricity cost: $79.48

Enter kwh during Off Peak period: 0

Process finished with exit code 0
```

Non-Functional Specifications:

1. The program should strictly adhere to the input and output requirements, **particularly the order of the input and the labeling of the output cost**.

2. The program should include brief comments in your code identifying yourself, describing the program, and describing key portions of the code.

3. Assignments are to be done individually and **must be your own work**. Software may be used to detect academic dishonesty (cheating).

4. Use Python coding conventions and good programming techniques. For example:
   a. Meaningful variable names
   b. Conventions for naming variables and constants
   c. Use of constants where appropriate
   d. Readability, indentation, and consistency

5. The name of the file you will submit must be your **Assign1.py**. ***Make sure you attach your Python source file to your assignment submission; do not put the code inline in the textbox. Asssignment submission can be found under* Assignment 1 *in the* Assignments Tab *in the OWL course site.***

   Make sure that you develop your code with Python 3.9 as the interpreter and that you have executed it with PyCharm EDU 2021.1; failure to do so may result in the testing program failing (see below).

Marking of the Assignment:

1. **Your program will be executed by an automated testing program.** This testing program assumes that:
   a. The program name is **Assign1.py** .
   b. That you are using Python 3.9.
   c. That you have submitted it via OWL by uploading it.
   d. That you have adhered to the input/output specifications.

   Failure to adhere to these constraints will likely cause the testing program to fail. This may require a remarking of your program which will include a 10% penalty.

2. Functional specifications:
   a. Does the program behave according to the specifications found in the assignment document?
   b. Does the program handle input?
   c. Is the output according to specifications?

3. Non-functional specifications as described above.