

Table of Contents

- Background
- Problem statement
- Instructions
 - electricity.py (main program)
 - energy_tier() function
 - energy_cost() function
 - main() function
 - Other instructions
- test_electricity.py (test script)
 - Tests of energy_tier()
 - Tests of energy_cost()
- Testing your code
 - Running your program at the command line
 - Using your test script
- Submitting your work
- Academic integrity

Homework: Cost of electricity

Background

The concept for this homework comes from Shenghan Gao. Many thanks!

Many utility companies charge for electricity based on the time of day when the electricity is consumed (among other factors). For this homework, we'll use a three-tiered pricing scheme based loosely on the one described [here](#). The three tiers and corresponding prices we will use are as follows:

Tier number	Tier name	Price of electricity
0	Super off peak	\$0.09 per kilowatt-hour
1	Off peak	\$0.10 per kilowatt-hour
2	Peak	\$0.17 per kilowatt-hour

The tiers correspond to the following times of day (times expressed in 24-hour time):

Time range	Tier
0:00–4:59	Super off peak
5:00–5:59	Off peak
6:00–8:59	Peak
9:00–16:59	Off peak
17:00–19:59	Peak
20:00–23:59	Off peak

For this assignment, you will write a program that reads in a comma-separated value (CSV) file containing a record of a customer's electricity usage over a period of time (usually a month). Each line in the file will contain an hour (expressed in 24-hour time) and a number of kilowatt-hours (kWh) of electricity used in that hour. The two values will be separated by a comma. Below is a snippet of an example file:

```
10,1.24
11,1.14
12,0.77
13,1.11
14,1.26
15,0.66
```

Problem statement

- Write a program that will read in a CSV file containing a customer's electricity usage over a period of time and output the cost of the user's electricity based on the pricing scheme specified in the [Background](#) section.
- Write a test script to test certain functions from your program.

Instructions

electricity.py (main program)

Use the supplied template `electricity.py` to build your solution. Define your functions after the `import` statements and before the `parse_args()` function (you may delete the comment that serves as a placeholder for your functions). You should develop your program in VS Code. Your program should be named `electricity.py`.

energy_tier() function

Write a function called `energy_tier()`. This function should take one argument: an hour, specified as an integer between 0 (12 midnight) and 23 (11pm).

Your function should return an integer:

- 0 if the specified hour falls within a super off peak time range
- 1 if the specified hour falls within an off peak time range
- 2 if the specified hour falls within a peak time range

energy_cost() function

Write a function called `energy_cost()`. The function should take two arguments:

- a tier (an integer between 0 and 2, inclusive, as described under `energy_tier()` above)
- an amount of electricity used (a float), in kWh

Your function should return the cost of the energy used. This will be the product of the number of kWh of electricity used and the price of a kWh of electricity for the specified tier (see the price table in the [Background](#) section). For example, if the specified tier were 2 (peak) and the number of kWh used was 2.0, the function should return 0.34 (which is $0.17 * 2.0$, where 0.17 is the cost of peak electricity, and 2.0 is the number of kWh of electricity used).

main() function

Write a function called `main()`. The function should take one argument: a string containing the path to a CSV file as described in the [Background](#) section.

This function should open the specified file for reading and loop over the lines of the file. It should calculate the total cost of the electricity use reported in the file. It should return this amount as a float.

Other instructions

- Please write docstrings for each of your functions. Your docstrings should start with a brief statement of the function's purpose. Include an "Args:" section to document any arguments, and a "Returns:" section to describe the return value (if any). Docstrings were covered in the lectures here: <https://youtu.be/jHTv83PlQYw?t=1415>. There's an ELMS page about them here: <https://umd.instructure.com/courses/1299872/pages/docstrings>.
- Please keep your lines of code to 80 characters or less. If you need help breaking up long lines of code, please see <https://umd.instructure.com/courses/1299872/pages/how-to-break-up-long-lines-of-code>.

test_electricity.py (test script)

Create a Pytest test script to test the `energy_tier()` and `energy_cost()` functions. Name your test script `test_electricity.py`.

At the top of your test script, import your program (or import the functions from your program to be tested). Also import code that will allow you to accurately test floating-point values (for example, `math.isclose()` or `pytest.approx()`).

Tests of energy_tier()

Write a test function to test the `energy_tier()` function. Add test cases for the following times:

- 0:00
- 5:00
- 6:00
- 9:00
- 17:00
- 20:00

Add test cases for three other times as well.

Tests of energy_cost()

Write a test function to test the `energy_cost()` function. Add test cases for each tier, using 1 (i.e., 1 kWh) for the second argument to `energy_cost()`. These tests will ensure that your function returns the correct cost per kWh.

Add at least one more test case per tier where the second argument is not 1, to ensure that the math in `energy_cost()` is correct.

Be sure to use a test method that will allow you to account for the kind of small inaccuracies that often arise when dealing with floating-point numbers.

Testing your code

Running your program at the command line

The template is designed to use a command-line argument. To run your program within the VS Code built-in terminal, first make sure you have opened (in VS Code) the folder where your program is saved. If necessary, you can go to the VS Code File menu and select "Open..." on macOS or "Open Folder..." on Windows, and navigate to the directory where your program is.

Then, open the VS Code built-in terminal. Type `python3` (on macOS) or `python` (on Windows) followed by a space, the name of the program, another space, and the name of the CSV file containing a customer's electricity usage over time (a sample file, `electricity_use.csv`, is provided with the assignment). Below is an example. The example assumes that `electricity_use.csv` is in the same directory as your program.

```
python3 electricity.py electricity_use.csv
```

Using your test script

Your test script should be in the same directory as your main program. To run your test script within the VS Code terminal, first make sure you have opened VS Code to the folder where your program is saved. Then open the terminal and type the following at the command line:

```
pytest test_electricity.py
```

If that does not work, try

```
python3 -m pytest test_frequency.py
```

Submitting your work

Separate Gradescope assignments are provided for the two files you will create (`electricity.py` and `test_electricity.py`). Please upload each file to the correct Gradescope assignment. In case of questions, please contact a member of the instructional team.

`electricity.py` will be partially auto-graded by Gradescope. If you are not happy with the results, you may revise your code and resubmit.

Academic integrity

This assignment is to be done by you individually, without outside help of any kind (including, but not limited to, help from classmates, tutors, or the internet). Disseminating these instructions in whole or in part without written permission of the

Table of Contents

- Background
- Problem statement
- Instructions
 - electricity.py (main program)
 - energy_tier() function
 - energy_cost() function
 - main() function
 - Other instructions
- test_electricity.py (test script)
 - Tests of energy_tier()
 - Tests of energy_cost()
- Testing your code
 - Running your program at the command line
 - Using your test script
- Submitting your work
- Academic integrity

instructor is considered an infraction of academic integrity.

Last updated 2021-02-15 12:11:30 -0500