
Programming Problem 1: sprinkler.py

Directions: Download the template files I have provided on BrightSpace. Then open Spyder, load these template files, and write the following programs. Submit your source code via Gradescope in .py format. READ THE INSTRUCTIONS on submitting your work in the Course Documents section of BrightSpace.

Specify collaborators/resources or **explicitly specify that none were used** in the comments at the top of your .py file. You need not list me or our class notes as collaborators/resources. **Failure to include this will result in a zero on the assignment.**

Be sure to read the SPECIFICATIONS carefully! And write comments!

Assignment goals: Get practice with various numerical operations in Python, getting user input, and appropriately formatting output.

You are working for a gardening company. You will write a program that asks the user for information about their house — namely, the length and width of the yard (both in feet) — and then compute the amount of sprinklers and amount of water used to water their yard.

Here is a sample interaction of the way that your program will work (the numbers 30 and 40 are entered by the user; the program should produce the rest):

```
Enter width (in ft): 30
Enter length (in ft): 40
You have 1200.0 square feet of yard area and 960.0 square feet for irrigation.
You will need 7 sprinklers in your yard.
It will use about 21.77 gallons per minute when running.
Your bill will be about $58.81 per month.
```

To perform the calculations shown in this sample, we will make some highly simplifying assumptions about the water usage and sprinkler ranges — but they are not too far off (see <https://redfordsupply.com/blogs/news/how-many-square-feet-does-one-irrigation-zone-cover>).

Here are the assumptions used to calculate the quantities found in the sample output above:

- To calculate the usable area for irrigation, we will assume exactly **80%** of the yard is needing irrigation and the other half is used for a deck.
- A single sprinkler waters **155 ft²** of the yard. The number of sprinklers that you need for the yard assumes perfect alignment, for example, 1000 ft² of usable area is $1000/155 = 6.45...$ rounded up, in other words, 7 irrigation (to accomplish this, you can use the `int()` or the `math.ceil()` functions).
- A single sprinkler is rated for **3.11 gallons per minute** and it is recommended to run an irrigation system once per day for 15 minutes. In New York, the water system costs **\$4.49 per 1000 cubic ft (aka 748 gallons)**. You may assume that there are always 30 days in a month.

Fill in `sprinkler.py` with a program that performs as specified here.

Hint: make sure you try calculating my sample values by hand before programming to ensure you understand the task!

The output should be *exactly* as specified in the sample above. When I say exact, I mean identical, down to the spacing. In this class, I have written autograders for most assignments, which run on your code as soon as you submit it and let you know if you pass the tests I've written. Those tests check if your output

matches what I've specified down to the spaces, so you must follow my specifications exactly. (This may seem annoying, but it is a valuable skill to master. It is very common in computer science to have precise instructions for something to work or for a client to be happy.) The benefit of autograders is that you can immediately know if your code isn't doing exactly the right thing and fix it before the deadline!

Specifications: your program must

- ask the user to enter the width and length of their yard in feet (as formatted in the sample run above)
- calculate and output the following quantities, as formatted in the sample run above:
 - the total area (in square feet) of the yard and the area (in square feet) usable for the irrigation system.
 - the number of sprinklers that can fit in the yard (on the usable area)
 - the number of gallons per minute the sprinkler system will use
 - the overall cost per month of using the sprinkler system.

Finally, any program with more than a few lines of code (so nearly all of them) should have comments describing what blocks of the code do. We'll learn more about how to organize code better in the future, but for now, you should put in a couple of comments describing what blocks of code are doing so that someone reading your program can easily see what is going on. You should also ensure any variables you use are named well, and your header describes what the program does. Finally, be mindful of the spacing in your program: how can you use an extra line here or there to make it easier to read, but not too many? There are several correct answers to these style questions, so go with what you think makes it easiest to read and stay consistent. I will give feedback on your style to help you improve it.

Note: This problem has two visible test cases on Gradescope. You will see if your code passes these two test cases. In addition, it has two invisible test cases, which you will not be able to see until after grades are released (you will not see if your code passed or did not pass these). So, run additional tests to cover other possible inputs! (You must work out the correct output in your test cases. You may get inaccurate answers in some cases due to "float weirdness" – don't worry about trying to fix that.)

Again, your input prompts must match the ones given above *exactly* (including spaces) to pass the test cases on Gradescope.

You can submit your code as many times as you wish before the deadline to fix any issues. If you have questions about what the autograder tells you is wrong, please ask!