Assignment 1 Python - Customizing House Plans

Due Jan 21, 2016 by 12:30pm **Points** 80 **Submitting** an external tool
Available until Jan 23, 2016 at 12pm

This assignment was locked Jan 23, 2016 at 12pm.

Answer the code questions in a single Python file. Use what's you've learned so far about good style and documentation. Start by editing the following file:

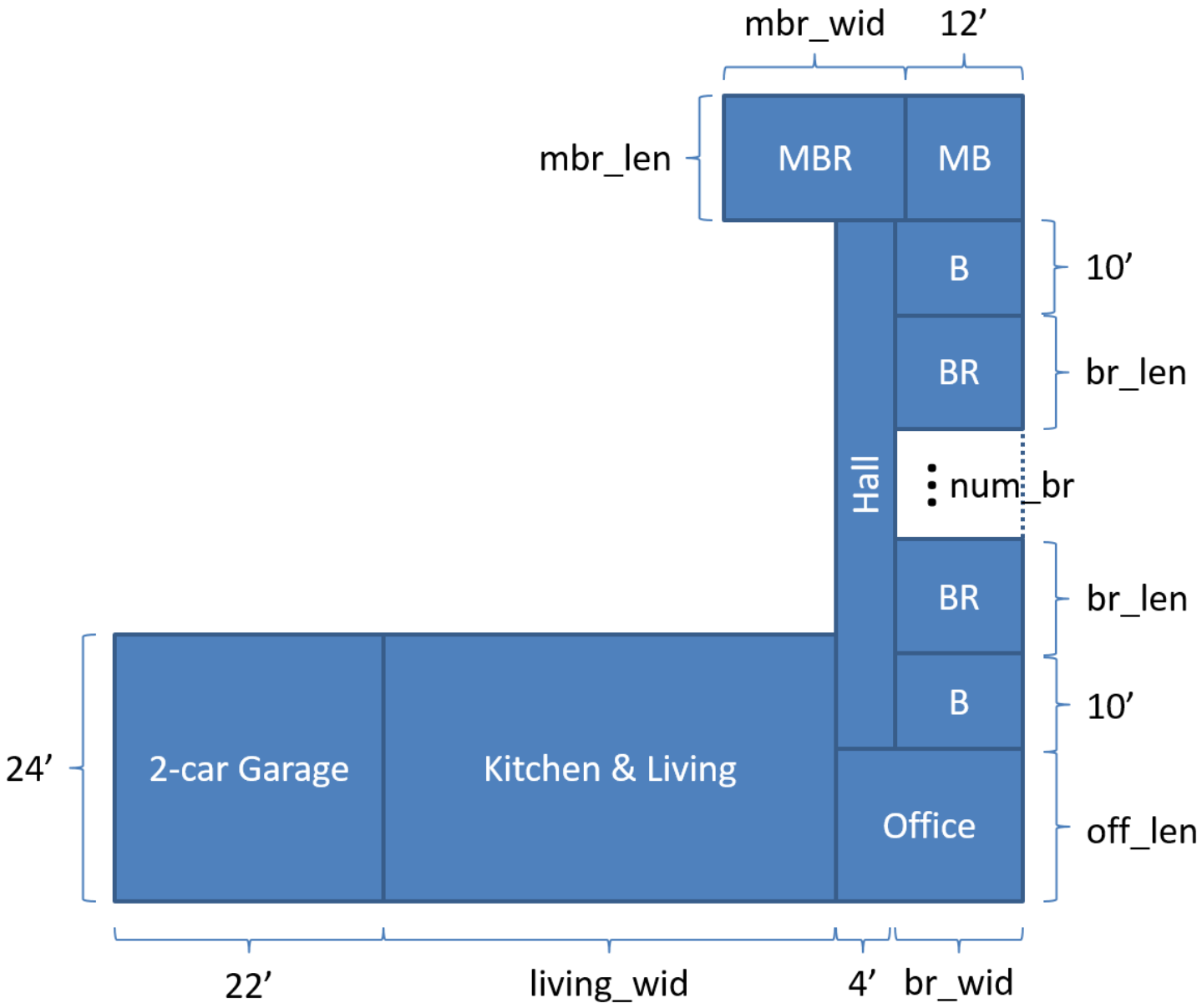
- **http://rice.codeskulptor.org/#comp130_assignment_houseplan.py (http://rice.codeskulptor.org/#comp130_assignment_houseplan.py)**
- **Test (http://codeskulptor.appspot.com/owltest/?urlTests=comp130.assignment_houseplan_tests.py&urlPylintConfig=comp130.pylint_config.py)**
- See submission instructions at the bottom of the page.

Big Picture

So far, you've been learning and practicing two main skills that you will use here on a larger, real-world (albeit simplified) problem.

- creating new functions based upon built-in mathematical operations, and
- breaking larger problems into smaller problems and creating new functions for each of these.

We run a small home-building firm. Calculating our firm's cost and our price to the customer can take a lot of effort and time, since each house is different. Not only can the house plan vary, so can all of the finishes, such as appliances, windows, and flooring. We want a program to provide a quick price estimate during customer meetings. At least for now, we'll be satisfied with a program that only works for our most popular customizable house plan. This program will calculate a price estimate based upon some of the customizable dimensions, but will assume a typical level of finish.



The above diagram is the house plan you will work with. It consists of three "wings". On the bottom is the public wing, with a garage and combined kitchen and living space, which has a customizable width. On the right is a bedroom wing, with an office, hallway, two bathrooms, and a series of identical bedrooms. Both the length and number of these bedrooms is customizable. Finally, at the top is the master bedroom wing, with a bedroom and bathroom. The master bedroom is customizable in both width and length. More specifically,

- The combined kitchen/living area size is given as `living_wid` × 24 feet.
- The master bedroom size is given as `mbr_wid` × `mbr_len`.
- There are a total of `num_br` bedrooms other than the master bedroom. While the diagram shows two separated by ellipses, `num_br` can be zero or more. Each of these bedrooms is of the same size, given as `br_wid` × `br_len`.
- The is 4 feet wide, and its length depends on `num_br` and `br_len`.
- There are exactly three bathrooms — the master bathroom and two other identical bathrooms.
- The office length is given as `off_len`.

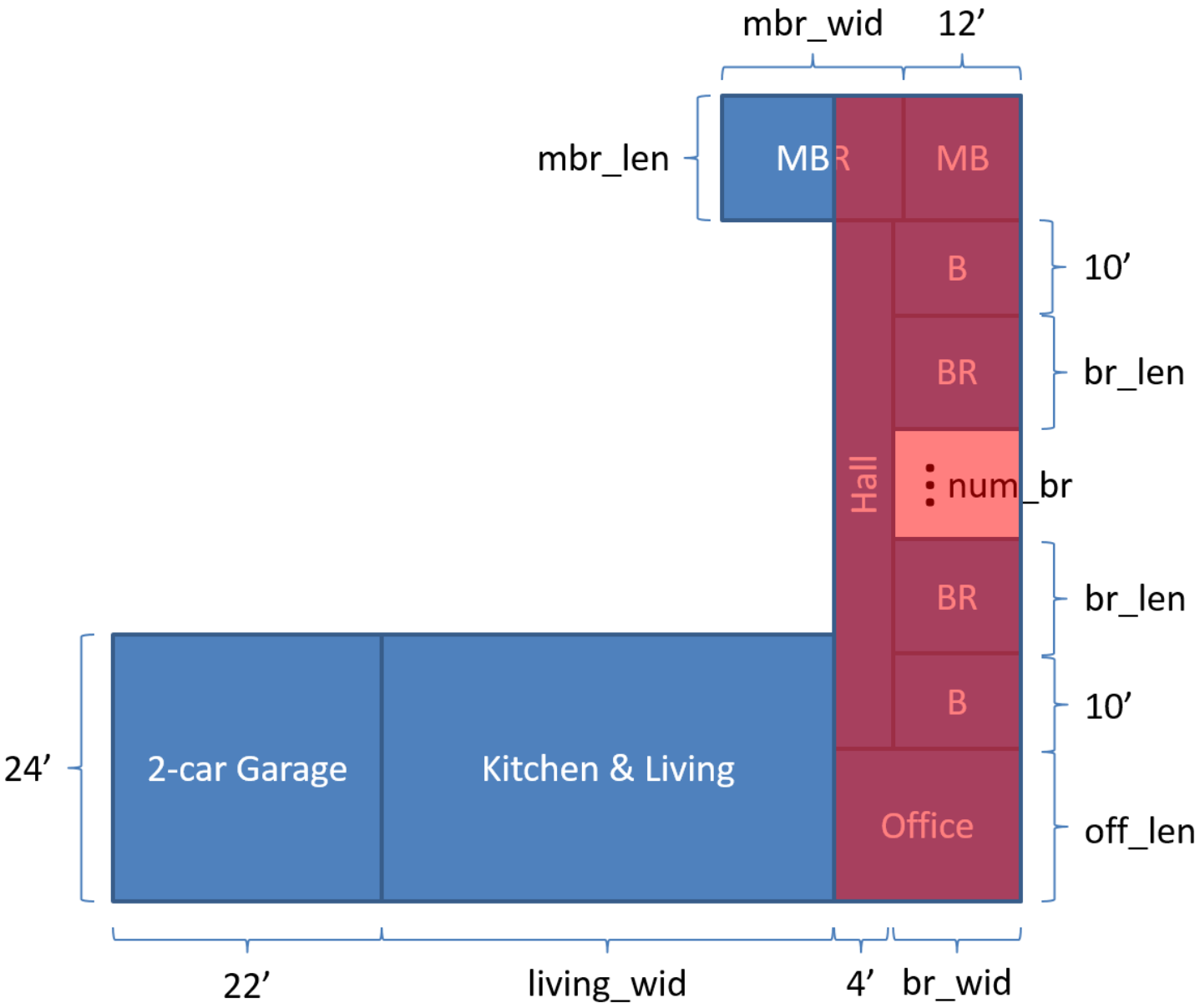
The remaining dimensions are either fixed or calculated from the above as depicted in the picture above.

Your program will compute a cost estimate based upon a combination of the square footage and the perimeter length. Each square foot is $100, while each perimeter foot averages an additional $80.

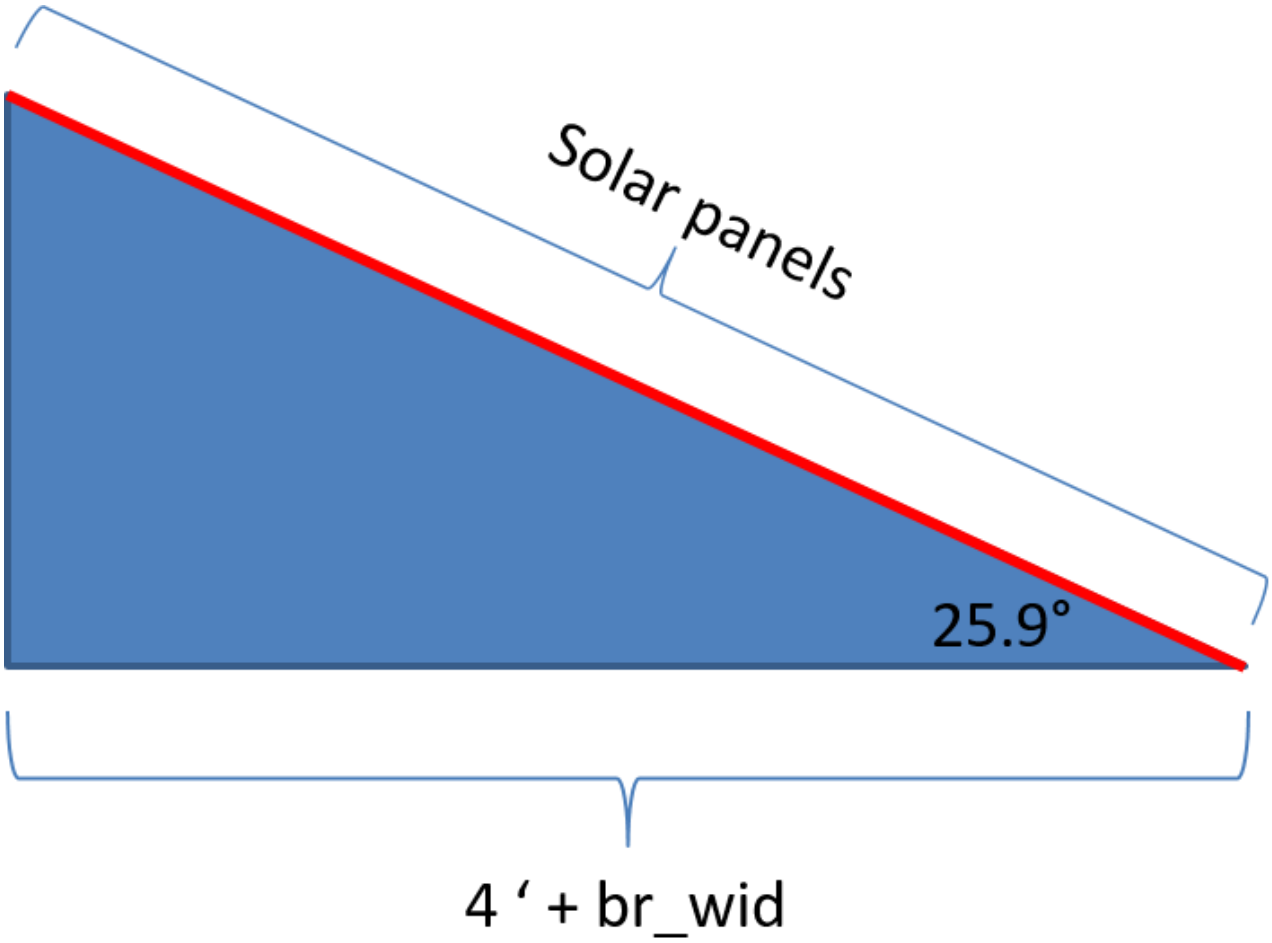
Additional details: Your code may assume that all of the dimensions are positive numbers, and that the number of extra bedrooms is a non-negative integer. Also, you can assume the length of the bedroom wing is sufficient so that the living area doesn't touch or overlap with the master bedroom. Finally, you can assume the master bedroom and bathroom are together at least as wide as the hallway and other bedrooms. In short, your code will not be given any funky cases that don't look like the picture.

In addition, our company always installs solar panels on the roof of the hallway/bedroom wing. To maximize the space for these panels, this wing uses a *shed*-style roof, with the peak at the left, instead of the more traditional *gable*-style, which would peak in the middle. The panels are installed from the outside wall of the office to the outside wall of the master bedroom, in the area shaded in red in the next pictures. (More realistically, the roof would extend past the walls, and the panels wouldn't reach to the edges of the roof, but we'll assume these two details cancel each other exactly.)

View from above:



View from front of house, only showing bedroom wing from its side:



The cost of these solar panels is already included in the previous calculation. Now we want to calculate the power generated from the solar panels. For this, we need to make a few assumptions, which would appear as constants in your program:

- The house is located in Houston (i.e., 30 degrees latitude) and oriented so that the panels face true south. The given roof angle is optimized for Houston. This gives an *insolation* of 6.1 kilowatt hours per square meter per day. (Cite: **www.solarpaneltilt.com** **(http://www.solarpaneltilt.com)**)
- The solar panels are 22% efficient, i.e., converting roughly a fifth of the solar energy into electricity. (Cite: **a recent comparison of commercially available panels** **(http://www.greentechmedia.com/articles/read/Worlds-Most-Efficient-Rooftop-Solar-Panel-Revisited)**)

You will write the function `solar(off_len, num_br, br_wid, br_len, mbr_len)` to calculate the power generated (in kilowatt hours per day) from these solar panels. The basic calculation you need is *power* = *area* × *insolation* × *efficiency*. However, note that our house dimensions are in feet, while meters are standard in this calculation. 1 foot = 0.3048 meters.

As a mathematical hint, you'll need a bit of trigonometry.

Grading

- (60 points) `cost()` correctness.
- (10 points) `solar()` correctness
- (10 points) Code style -- See course **code style guidelines**.

An Aside

It is certainly annoying on this assignment that the functions have **many** input parameters. There must be a better way. Indeed, it would be better to package these multiple parameters into some notion of a single "house design", but we haven't learned how to do that yet. Later in the course, we can do that with the simple notion of a "tuple" or the more complicated "class".

Submission