Lab 9: Making a position class and setting up pygame

Part 1: Making a position class

A position in two dimensions is a set of coordinates x and y. For this lab, you will write a position class with the following methods:

- `__init__`, which takes parameters self, x, and y, and sets the initial postion (`self.x` and `self.y` are set to x and y).
- `add_other`, which adds a different position to the current position and stores the result in self. So, if `A` and `B` are positions, `A.add_other(B)` will change `A` by adding on position `B`. To add positions, add up the x coordinates and the y coordinates.
- `up`, which moves the position up by adding 1 to `self.x`. For this lab, you don't need matching down, left and right methods, but it would be a nice touch.
- `print`, which prints out the position in this format: `(5,2)`

Here is some code to test your position class (posted as lab9.py):

```
A = Position(1, 1)
B = Position(2, 3)
A.print()
B.print()
A.add_other(B)
A.print()
A.up()
A.print()
C = Position(-3, -8)
A.add_other(C)
A.print()
```

Here is the expected output:

```
(1,1)
(2,3)
(3,4)
(4,4)
(1,-4)
```

Part 2: Setting up pygame

Pygame can be downloaded from http://www.pygame.org. Newer versions may be available from: https://bitbucket.org/pygame/pygame

Installation instructions are at pygame.org for all major platforms. If you are using the lab computers, pygame is not presently installed. It should not be hard to install python on this computers, but if you run into a problem, the simplest solution is probably to install portable python: http://portablepython.com

Once you have pygame installed, try to run the program lab9_pygame_test.py from the lab section of the course website. Download ball.bmp into the same folder as lab8_pygame_text.py before you try to run it. If things are working correctly, you will see a window appear with a ball moving up and down. There is no programming required for this part of the lab, but having pygame working correctly will be important for project 1.