Python In-class Programming Assignment
Dictionaries Of Objects

In this exercise, we're going to create a dictionary whose keys are strings and whose values are objects of a class you will write. We'll also place our class definition in a separate file and import it as a library.

The data file we're examining contains population data for various countries based on counts taken in 2007, 2008, 2009, and 2010. Each line of the file is composed of the following items, separated by a comma[1]:

```
countryName, pop2007, pop2008, pop2009, pop2010
```

Make sure your repos are up to date. The `~/repo201/classwork/cw47` directory contains several files you'll use for this assignment. Copy the entire contents of that directory to a location of your choosing, using the following commands in a terminal window:

`cd ~/Desktop` (*or to a work space of your choosing*)
`cp -R ~/repo201/classwork/cw47/* .` <- *The trailing space and period are important!*

Perform the following steps:

1. Open the `classes.py` file in Atom. The `classes.py` file is located in the `utils` directory that you just copied to your workspace.

2. Finish writing the definition for the `Country` class. The `Country` class contains one initializer and five *getters*. The initializer should break a single line from the file into its component parts and assign those parts to the associated properties. A line (represented as a string) is passed to the initializer via the `line` parameter. The `Country` class has five private properties: `__countryName`, `__pop2007`, `__pop2008`, `__pop2009`, `__pop2010`. *Note: Country names can stay as strings, but population data should be cast to `float`!*

3. Open the `countryStats.py` file in Atom. This is the starter code for the main program. Complete the code required to build a dictionary of all the country data, where each *key* in the dictionary is the name of a country (string), and each *value* in the dictionary is a `Country` class object.

4. Complete the `main()` function to allow the user to enter a single letter (`a -> z`), either uppercase or lowercase. The program should then print all the countries that start with that letter, sorted alphabetically, along with the population data for each country. Keep

[1] Data collected from *https://www.data.gov/*. All population numbers represent the country's population in millions.

asking for a single letter until the user enters a blank line (enter by itself).

5. An executable version of a working program is included in this directory so you can see how your program is supposed to work. It's called `countryStats`. Assuming you've copied all the files from `cw47` as described earlier, you can run `countryStats` by first adjusting its permissions with the following command: `chmod 755 countryStats`. You only have to perform this step once. After that, you can run the program any time you want by typing: `./countryStats`.