Geography 378: Introduction to Geocomputing
Lab 4: Python Classes and Plotting

Assigned: 10/25
Due: 11/8
15 points

Hand-in

- Please collect your answers in a single .py file called **lab4_yourname.py.**
- Submit the file to the assignment folder called "Lab 4".
- Include appropriate comments to explain what each line or block of code accomplishes. **You must comment your code for full credit**.

Notes

- Under no circumstances should your program crash (i.e. it either restarts or ends gracefully).

Lab Task

1. (5 pts) In last lab, we created python scripts that read in the content of *CityPop.csv* and store the data in certain containers. Now, let's try to replace your containers with **classes** and create **instances** to store the data.
   - The name of your class should be `City`.
   - Your class should have an `__init__` method to assign values to the following attributes: city name, city label, latitude, longitude, population values from 1970 to 2010 (consider to store them in a dictionary)x.
   - Create a list called `Cities` to store the city instances based on reading the entries in CityPop.csv.
   - Print out the attributes of all cities at the end. (Note: `print Cities` won't work!)
   - Like last lab, you need to deal with bad inputs when trying to read the file.

2. (5 pts) Add the following **methods** to your City class to make it more useful. You don't need to get user input for this task. To test your methods, just call them using any valid input (from any valid city instance).
   - printDistance(self, othercity): calculate and output the distance between this city and a given city.
   - printPopChange(year1, year2): calculate and output the population change over two years. Note that *year1* and *year2* are two years (e.g. 'yr1990') instead of the population.

3. (5 pts) In the same script, choose your favorite two cities from the CityPop.csv and plot their population changes. You can ignore the zeros in the original table.