CIS 1250 Python Program7 – Create a class

# Turn in Requirements:

5 pts. Name your project LastnameP7, such as GarnerP7.

# Program Requirements:

1. 5 pts. Write the file name, your name, email address and purpose of the program at the top of your library source code in a comment.

# GarnerP7

# Programmer: Rob Garner

# EMail: Rgarner7 @cnm.edu

# Purpose: demonstrate how to define a class

1. 5 pts. Add comments as appropriate. Be sure that your program output is neatly presented to the user. Add documentation comments to your functions.
2. You are going to define a class to keep track of a point in space, it’s description and provide it the capability to figure out it’s distance from another point. You may be able to re-use some of the code from your previous work.
3. In your program define a class called GeoPoint that will have the following:
   1. An init (self) method that will set two attributes (variables) called self.lat, self.lon for the location of that point. The init method will also initialize a description attribute (variable) that will start off as the empty string “”.
   2. A SetPoint(self, lat, lon) method that will set the values of self.lat, self.lon
   3. A GetPoint(self) method that will return a tuple or list with self.lat, self.lon.
   4. A Distance(self, lat, lon) method that will figure out the distance between the object’s self.lat, self.lon and lat, lon parameters passed in.
   5. A SetDescription(self, description) method that will set the objects self.description attribute (variable).
   6. A GetDescription(self) method that will return the objects self.description attribute.
4. In the main part of your program do the following:
   1. Instantiate two points
   2. Use the SetPoint and SetDescription methods to set each of the points locations and descriptions. Make sure they have different coordinates and different descriptions.
   3. Inside a “Do another (y/n)?” loop do the following:
      1. Ask the user for their location. You can ask for coordinates in three inputs or ask them for their coordinates in one input with each element separated by a coma.
      2. Use point1 and point2’s Distance method to find the distance from each point to the user’s location

distanceToOne = point1.Distance(lat, lon)

distanceToTwo = point2.Distance(lat, lon)

* + 1. Tell the user which point they are closest to in this format:

You are closest to <description> which is located at <point’s coordinates>

* + 1. Ask “Do another (y/n)?” and loop if they respond with ‘y’