SelfEvaluation12_3

November 15, 2015

1 Exercise 12.3:

Modify the City class `__init__()` to test if the input lon and lat values are valid. If not the function should assign the default value.

Answer appears after one blank page (so you don't peek).

Are you sure you're ready to peek?

```
In [17]: class City:
           def __init__(self, name, lon=-999, lat =-999):
             # Check the values, we can allow the lon to be in either -180 - 180 or
             #  or 0 - 360.  It's up to the user to decide here.
             self.lon = lon
             self.lat = lat

             if not (-180 <= lon <= 360):
                 self.lon = -999
             if not (-90 <= lat <= 90):
                 self.lat = -999
             self.name = name
           def __repr__(self):
             return "<City: %s at (%.2f, %.2f)>" % (self.name, self.lon, self.lat)
           def __str__(self):
             return "From str method of City: %s is at lon:%.2f lat:%.2f" % (self.lon, self.lat)

           def printLatitude(self):
             print "%s is at latitude %.2f" %(self.name,self.lat)
           def printLongitude(self):
             print "%s is at longitude %.2f" %(self.name,self.lon)
           def printLatDiff(self, othercity):
             dist   = abs(self.lat - othercity.lat)
             format = "%s is %.1f degrees Latitude away from %s"
             values = (self.name, dist, othercity.name)
             print format % values

         # Expect lat to be -999
         test = City(name = 'town', lon = -91, lat = 180)
         print repr(test)

         # Should work:
         test = City(name = 'town', lon = 91, lat = -89)
         print repr(test)

         # Expect lon to work:
         test = City(name = 'town', lon = 359, lat = -89)
         print repr(test)
```

```
<City: town at (-91.00, -999.00)>
<City: town at (91.00, -89.00)>
<City: town at (359.00, -89.00)>
```