About Lab 11

Lab 11 gives you an opportunity to design classes on your own. We tell you what methods the class should have; the code is up to you. This should be a nice review of many of the things we have done this semester.

The two classes you need to create are Trip and RoundTrip. Both should be defined in a module Trips.py. You will save some code if you make RoundTrip be a subclass of Trip, but you need to be careful because some methods may work differently between the two classes..

Here are the methods you need to implement:

Constructors: The Trip constructor takes any number of arguments, including possibly no arguments. The first argument, if there is one, is the **source** or starting location of the trip. The remaining arguments are destinations to visit. The trip this represents starts at the source and visits each of the destinations in order. For example, Trip("Oberlin+OH", "Washington+DC", "New+York+NY") represents a trip that starts in Oberlin, proceeds to Washington, and then travels to New York. The RoundTrip constructor is similar, only a Roundtrip returns to its source after the last destination.

SetSource(city): This installs the given city as the trip source, overwriting any previous source. The destinations are unaffected.

GetSource( ): This returns a string: the source city if there is one or the empty string if there is no source

AddDestination (city): This adds the given city to the end of the list of destinations. For example, if a RoundTrip is currently from Oberlin to New York (and back to Oberlin) and we add Boston, then the new RoundTrip will proceed from Oberlin to New York to Boston (and back to Oberlin). This has no effect on the source of a trip.

InsertDestination(city, n): This adds the given city to the list of destinations, inserting it at position n. The destinations are numbered starting with 1. If a Trip currently has k destinations the valid positions for an insert are 1,2,...,k+1. If the trip currently has k destinations inserting at position k+1 has the same effect as doing an AddDestination. If we have a current Trip from Oberlin to New York and add Washington at position 1 the Trip will then proceed from Oberlin to Washington and then to New York.

RemoveDestination(city): This removes every instance of the given city from the list of destinations. The source is unaffected, even if it happens to be the given city. If this city is not one of the destinations a message to that effect should be printed.

GetDestinations( ): Returns a list of the destination cities.

Length( ): This returns the distance in miles for the complete Trip or RoundTrip. Remember that for RoundTrips this includes the distance from the final destination to the source.

Print( ): This prints a table listing each leg of the trip and the cumulative distance through that leg. For exampe, for the RoundTrip from Oberlin to Washington to New York to Boston to Buffalo this might print:

Oberlin+OH 0.00
Washington+DC 390.50
New+York+NY 615.41
Boston+MA 832.36
Buffalo+NY 1287.47
Oberlin+OOH 1510.58

Note that we give you an example program with a distance function that takes two cities as argumets and queries the Google Maps API for the distance between two cities. This example returns the number of meters between the cities but it should be easy to convert this into miles using the fact that there are 1609.34 meters in a mile. You should cut and paste this function into your code so you can call it for your Length( ) and Print( ) methods.

When your classes are complete an application program (in another file) like this shoud work:

```
import Trips

def main():
        r = Trips.RoundTrip("Oberlin+OH", "Washington+DC" )
        r.AddDestination("New+York+NY" )
        r.AddDestination("Buffalo+NY")
        r.InsertDestination("Boston+MA", 4 )
        r.Print()

main()
```