Python Tutorial II – Objects and Classes

Mark A. Austin

University of Maryland

austin@umd.edu
ENCE 201, Spring Semester 2025

August 29, 2025

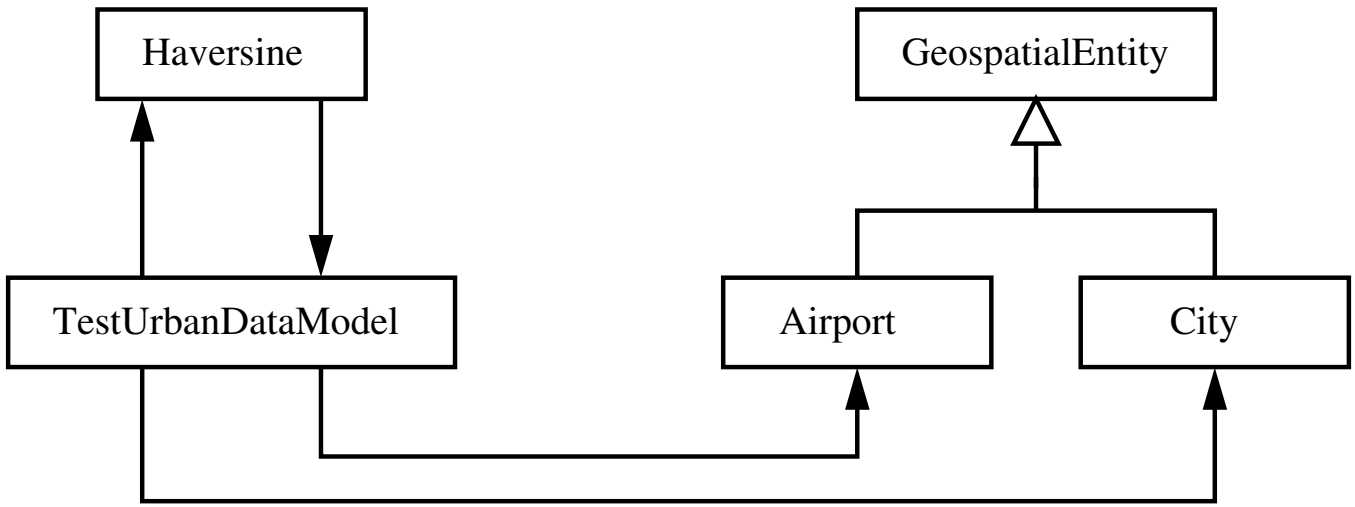
Overview

Case Study

(GeoModeling Spatial Entities)

Case Study: GeoModeling Spatial Entities

Geospatial Data Model: Create city and airport models. Use Haversine formula to compute distances between entities.



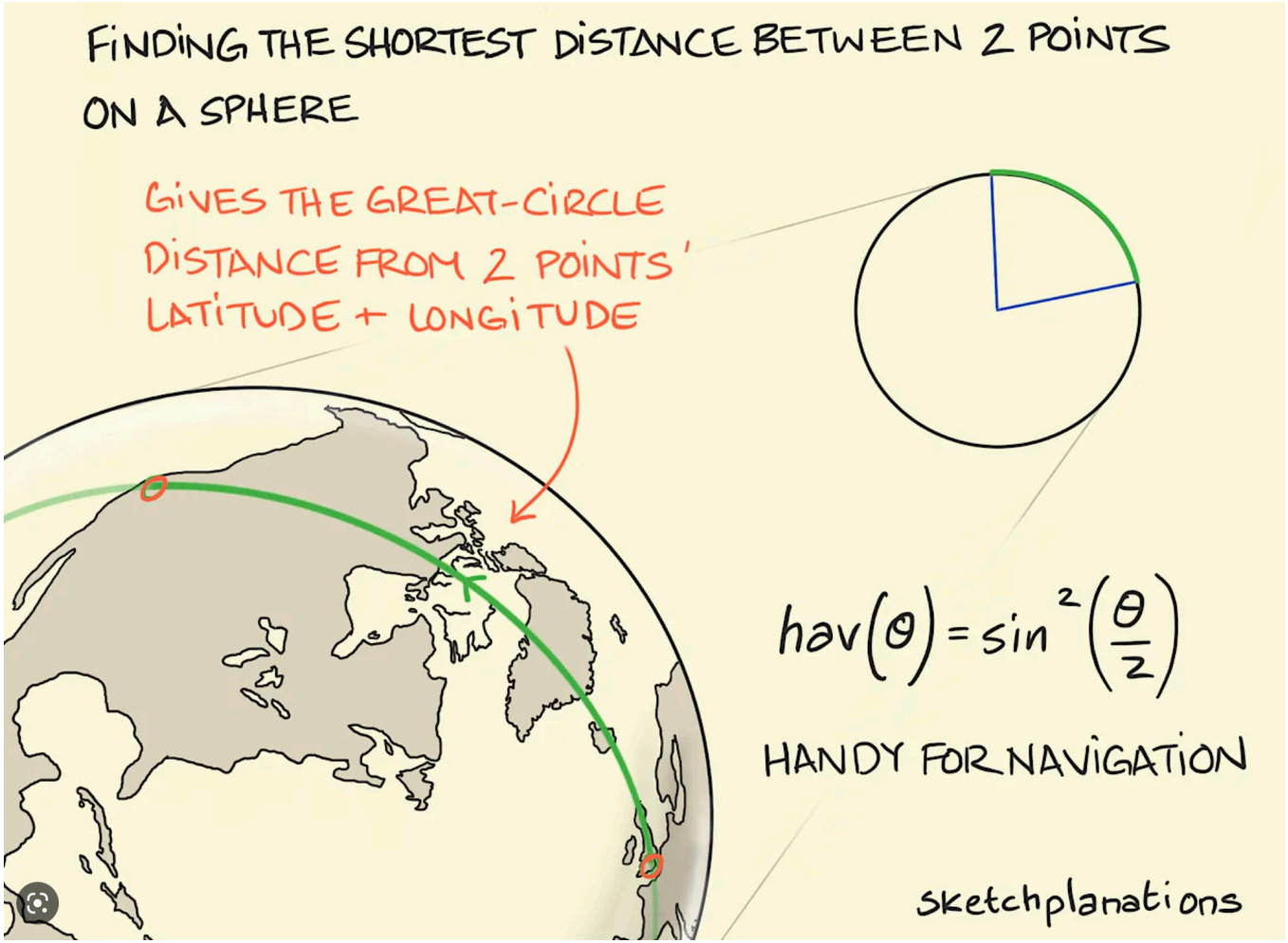
Geospatial Attributes: latitude, longitude, elevation.

City Attributes: name, population, state, country.

Airport Attributes: name, airport code.

Case Study: GeoModeling Spatial Entities

Haversine Formula



Case Study: GeoModeling Spatial Entities

Haversine Formula: Source code ...

```
1  # ==========================================================================
2  # Haversine.py. Small class that provides approximate distance (km) between
3  # two points using the Haversine formula.
4  #
5  # Call in a static context:
6  #
7  # Haversine.distance(47.6788206, -122.3271205,
8  #                    47.6788206, -122.5271205) --> 14.973190481586224 [km]
9  #
10 # earthRadius = 6372.8;     # Earth radius in KM
11 # earthRadius = 3959.87433 # Earth radius in miles.
12 #
13 # Written by: Jason Winn (http://jasonwinn.org)
14 # Modified by: Mark Austin                                    February 2023
15 # ==========================================================================
16
17 from math import radians, cos, sin, asin, sqrt
18
19 class Haversine:
20
21     # ====================================
22     # Compute haversine distance ...
23     # ====================================
24
25     @staticmethod
26     def distance(lat1, lon1, lat2, lon2):
```

Case Study: GeoModeling Spatial Entities

Haversine Formula: Source code ...

```
27          earthRadius = 3959.87433 # Earth radius in miles.
28          dLat = radians(lat2 - lat1)
29          dLon = radians(lon2 - lon1)
30          lat1 = radians(lat1)
31          lat2 = radians(lat2)
32
33          a = sin(dLat/2)**2 + cos(lat1)*cos(lat2)*sin(dLon/2)**2
34          c = 2*asin(sqrt(a))
35
36          return earthRadius * c
```

Source Code: See: python-code.d/geospatial/

Case Study: GeoModeling Spatial Entities

Compute Distance between Washington DC and NYC

```
1   # ============================================================
2   # TestHaversine.py: Small test program for haversine formula.
3   # ============================================================
4
5   from Haversine import Haversine
6   from City import City
7   from Airport import Airport
8
9   # main method ...
10
11  def main():
12      print("--- Enter TestHaversine.main()      ... ");
13      print("--- ============================== ... ");
14
15      print("--- Part 1: Create sample cities and airports ... ");
16
17      city01 = City( "Washington DC", 38.907192, -77.036871, 410.0,  5 )
18      city02 = City(     "Baltimore", 39.290385, -76.612189, 480.0, 10 )
19      city03 = City(      "New York", 40.712784, -74.005941, 265.0, 10 )
20
21      airport01 = Airport( "Baltimore-Washington", "BWI", 39.177404, -76.668392, 148.0 );
22      airport02 = Airport( "Washington Dulles",    "IAD", 38.952934, -77.447741, 313.0 );
23
24      print("--- Part 2: Print details of cities and airports ... ");
25
26      print(city01);    print(city02); print(city03)
```

Case Study: GeoModeling Spatial Entities

Compute Distance between Washington DC and NYC

```
27      print(airport01); print(airport02)
28
29      print("--- Part 3: Compute distances between locations ... ");
30
31      # Compute distance between Washington DC and Baltimore ...
32
33      lat1 = city01.getLatitude(); lon1 = city01.getLongitude()
34      lat2 = city02.getLatitude(); lon2 = city02.getLongitude()
35      d1 = Haversine.distance(lat1, lon1, lat2, lon2)
36
37      print("--- Distance: Washington DC to Baltimore --> {:f} miles ..".format(d1))
38
39      # Compute distance between Washington DC and New York ...
40
41      lat1 = city01.getLatitude(); lon1 = city01.getLongitude()
42      lat2 = city03.getLatitude(); lon2 = city03.getLongitude()
43
44      d1 = Haversine.distance(lat1, lon1, lat2, lon2)
45
46      print("--- Distance: Washington DC to New York --> {:f} miles ..".format(d1))
47
48      # Compute distance between IAD and BWI ...
49
50      lat01 = airport01.getLatitude(); lon01 = airport01.getLongitude()
51      lat02 = airport02.getLatitude(); lon02 = airport02.getLongitude()
52
53      d1 = Haversine.distance( lat01, lon01, lat02, lon02)
```

Case Study: GeoModeling Spatial Entities

Compute Distance between Washington DC and NYC

```
55
56          code01 = airport01.getAirportCode()
57          code02 = airport02.getAirportCode()
58          print("--- Distance: {:s} to {:s} --> {:f} miles ..".format( code01, code02, d1))
59
60          print("--- ============================== ... ");
61          print("--- Leave TestHaversine.main()      ... ");
62
63      # call the main method ...
64
65      main()
```

Source Code: See: python-code.d/geospatial/

Case Study: GeoModeling Spatial Entities

Abbreviated Output:

```
--- Enter TestHaversine.main()      ...
--- ============================== ...
--- Part 1: Create sample cities and airports ...
--- Part 2: Print details of cities and airports ...

--- City: Washington DC ...
--- ------------------------------------------------
---   Latitude    =    38.907192 ...
---   Longitude   =   -77.036871 ...
---   Elevation (highest) = 410.00 ft ...
---   Population =   5.00 ...
--- ------------------------------------------------

--- City: Baltimore ...
--- ------------------------------------------------
---   Latitude    =    39.290385 ...
---   Longitude   =   -76.612189 ...
---   Elevation (highest) = 480.00 ft ...
---   Population =  10.00 ...
--- ------------------------------------------------

--- City: New York ...
--- ------------------------------------------------
---   Latitude    =    40.712784 ...
---   Longitude   =   -74.005941 ...
---   Elevation (highest) = 265.00 ft ...
---   Population =  10.00 ...
--- ------------------------------------------------
```

Case Study: GeoModeling Spatial Entities

Abbreviated Output: (Continued) ...

```
--- Airport: Baltimore-Washington (BWI) ...
--- ------------------------------------------------
---   Latitude    =    39.177404 ...
---   Longitude   =   -76.668392 ...
---   Elevation (highest) = 148.00 ft ...
--- ------------------------------------------------

--- Airport: Washington Dulles (IAD) ...
--- ------------------------------------------------
---   Latitude    =    38.952934 ...
---   Longitude   =   -77.447741 ...
---   Elevation (highest) = 313.00 ft ...
--- ------------------------------------------------

--- Part 3: Compute distances between locations ...

--- Distance: Washington DC to Baltimore --> 34.931571 miles ..
--- Distance: Washington DC to New York --> 203.608912 miles ..
--- Distance: BWI to IAD --> 44.605415 miles ..

--- ============================== ...
--- Leave TestHaversine.main()      ...
```

References

- ....
- ....