xkdc: A webcomic of romance, sarcasm, math, and language.
OBJECT-ORIENTED COMPUTER PROGRAMMING FOR GIS

Lecture 0: Syllabus/Introduction

Dr. Giovanni Baiocchi¹

September 2014, College Park

¹Department of Geographical Sciences, UMD.
Outline

Course Overview
  Course Instructors
  Course goal
  Course objectives
  Software

Course Timetable

Assessment

Questions
Course Instructor, Teaching Assistant

- **Instructor**
  - Dr. Giovanni Baiocchi, Associate Professor at the Geography Department UMD
    - Email: baiocchi@umd.edu
    - Office hours: Fridays 2–3 pm, room 1133

- **Teaching assistant**
  - Patrick McDonough
    - Email: p.mcdonough8133@gmail.com
    - Office hours: Wednesday 12:30-2:00pm, Thursday 1:30–3:00pm, Teaching Assistant room
Course Aims

DEVELOP GENERAL COMPETENCY IN:

- Writing simple code using an Object-Oriented programming language
- Automating components of work in IPython
- Conducting customized analyses of geospatial data sources
Course objectives

▶ Develop a deeper understanding of programming techniques (applied to Python)
▶ Expand programming capabilities within GIS
▶ Develop proficiency in using a modern computational environment to automate processing of geospatial data
▶ Develop ability to problem-solve independently
▶ Introduce more advanced Python and geospatial Python-based open-source packages used to process geospatial information and rapidly expanding new GIS open-source data formats
▶ Acquire necessary information from help files
Course scope

- Object-oriented programming (Python and GIS packages)
- Developing custom applications for geospatial data manipulation and analysis
- Prerequisites: GEOG 373 or another GIS course, GEOG 376 (former GEOG398A) or another basic programming course
Main Software

- The required software for this class is **Python**. Python is the open source and freeware, one of the most powerful and versatile programming languages, and is available for free download for use on PC, Mac, UNIX and Linux environments.

- The recommended installation is **Anaconda**, available at [here](#).

- For Python I will use the **IPython notebook**, a web-based interactive and computational environment [here](#).

- If you install Anaconda’s Python distribution, available for most platforms, the notebook and most **useful packages** will be already installed, including the IPython Notebook (NumPy, SciPy, Pandas, IPython, Matplotlib).

- If you are already using **other Python** distribution I do not recommend installing Anaconda on top of it (unless are a consummate programmer!)

- Other **open-source GIS packages** needed can be installed on demand when required depending on the distribution and operating system you use.

- The software is available in the Lefrak/Geography **Open Lab** on the PC machines. If you have a laptop or home computer, get this software and download it immediately.

Object-oriented programming (Python and GIS packages)

**Dr. Giovanni Baiocchi**
Main Python Packages (already installed)

- **The IPython Notebook.** The IPython Notebook is a web-based interactive computational environment where you can combine code execution, text, mathematics, plots and rich media into a single document.

- **NumPy.** NumPy is the fundamental package for scientific computing with Python.

- **Python Data Analysis Library.** pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

- **matplotlib.** matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.
GIS-based Python Packages: might need separate installation

- **pyproj.** Performs cartographic transformations and geodetic computations.
- **fiona.** Fiona provides uncomplicated Python interfaces to functions in OGR, the best open source C/C++ library for reading and writing geographic vector data.
- **GDAL.** GDAL: Geospatial Data Abstraction Library
- **geopy.** geopy makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources
- **Shapely.** Shapely is a BSD-licensed Python package for manipulation and analysis of planar geometric objects. It is based on the widely deployed GEOS (the engine of PostGIS) and JTS (from which GEOS is ported) libraries.
- **PyShp.** This library reads and writes ESRI Shapefiles in pure Python. You can read and write shp, shx, and dbf files with all types of geometry. Everything in the public ESRI shapefile specification is implemented. This library is compatible with Python versions 2.4 to 3.x.
- **BaseMap.** Module for plotting data on maps with matplotlib
- **Descartes.** Use Shapely or GeoJSON-like geometric objects as matplotlib paths and patches
- **PySAL.** PySAL is an open source library of spatial analysis functions written in Python intended to support the development of high level applications.
- **GeoPandas.** GeoPandas is an open source project to make working with geospatial data in python easier. GeoPandas extends the datatypes used by pandas to allow spatial operations on geometric types. Geometric operations are performed by shapely. Geopandas further depends on fiona for file access and descartes and matplotlib for plotting.
- **Rtree.** Rtree is a ctypes Python wrapper of libspatialindex that provides a number of advanced spatial indexing features for the spatially curious Python user
- **folium.** Folium builds on the data wrangling strengths of the Python ecosystem and the mapping strengths of the Leaflet.js library
- **Mapnik.** Mapnik is a Free Toolkit for developing mapping applications. Above all Mapnik is about making beautiful maps.
- **Vincent.** The data capabilities of Python. The visualization capabilities of JavaScript.
Attendance policy, assignments, etc.

- **Attendance** is required for lectures, labs, mid-term, project presentations, and the final exams.
- Lab assignments are given on the day Labs are held and are due the next lab session, all due date modifications will be announced and posted on Blackboard, submission of assignments through course folder.
- Lab assignments and **collaboration**:
  - These are assignments to be completed individually!!!!
  - Program design and implementation MUST be completed independently
  - Minor help with debugging is OK.
Lecture structure

- One two-hour lecture per week (Tue 12:00 pm – 14:00 pm, room LEF 1124)
- 2 sections:
  - Concepts (lecture-like)
  - Implementation + exercises (hands-on) - you are encouraged to bring your laptop and practice writing code in the classroom
- I will NOT post lecture notes - be prepared to take very good notes during the class
Lab structure

- **One Lab per week**
  - Wednesday or Thursday 3 pm-5 pm, room LEF 1138

- **Lab preparation:**
  - Labs are focused on using Python for problem solving and are likely to take longer than 2 hours to complete if the subject is not examined before the lab
  - If applicable, come to the lab with most of your program pre-written in pseudo code that will speed things up
  - TA will address important issues and concepts at the beginning of each lab, come on time - he will not give a special encore presentation just for you
Recommended/Reference Textbooks

- Mark Lutz, Learning Python, 5th Edition
  - Get a comprehensive, in-depth introduction to the core Python language.
  - This self-paced tutorial gets you started with both Python 2.7 and 3.3- lines.
Recommended/Reference Textbooks

- Mark Lutz, Learning Python, 5th Edition
  - Get a comprehensive, in-depth introduction to the core Python language.
  - This self-paced tutorial gets you started with both Python 2.7 and 3.3- lines.

- Erik Westra, Python Geospatial Development - Second Edition
  - Build your own complete and sophisticated mapping applications in Python.
  - Walks you through the process of building your own online system for viewing and editing geospatial data
  - Practical, hands-on tutorial that teaches you all about geospatial development in Python
Recommended/Reference Textbooks

- **Mark Lutz, Learning Python, 5th Edition**
  - Get a comprehensive, in-depth introduction to the core Python language.
  - This self-paced tutorial gets you started with both Python 2.7 and 3.3- lines.

- **Erik Westra, Python Geospatial Development - Second Edition**
  - Build your own complete and sophisticated mapping applications in Python.
  - Walks you through the process of building your own online system for viewing and editing geospatial data.
  - Practical, hands-on tutorial that teaches you all about geospatial development in Python.

- **Joel Lawhead, Learning Geospatial Analysis with Python**
  - Construct applications for GIS development by exploiting Python.
  - Focuses on built-in Python modules and libraries compatible with the Python Packaging Index distribution system - no compiling of C libraries necessary.
Outline

Course Overview

Course Timetable

Assessment

Questions
Recommended/Reference Textbooks

<table>
<thead>
<tr>
<th>Week 1</th>
<th>Key Date</th>
<th>Event</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tue 02-Sep</td>
<td>12–14</td>
<td>Lecture 1</td>
<td>Introduction</td>
</tr>
<tr>
<td>Wed 03-Sep</td>
<td>15–17</td>
<td>LAB: 1</td>
<td>IPython Notebook</td>
</tr>
<tr>
<td>Week 2</td>
<td>Tue 09-Sep</td>
<td>12–14</td>
<td>Lecture 2</td>
</tr>
<tr>
<td>Wed 10-Sep</td>
<td>15–17</td>
<td>LAB: 2</td>
<td>Basic Python</td>
</tr>
</tbody>
</table>

**Advanced Python for GIS**

<table>
<thead>
<tr>
<th>Week 3</th>
<th>Key Date</th>
<th>Event</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tue 16-Sep</td>
<td>12–14</td>
<td>Lecture 3</td>
<td>Lists, Tuples, Dictionaries, and Sets</td>
</tr>
<tr>
<td>Wed 17-Sep</td>
<td>15–17</td>
<td>LAB: 3</td>
<td>Practice: Advanced Python Data Structures</td>
</tr>
<tr>
<td>Week 4</td>
<td>Tue 23-Sep</td>
<td>12–14</td>
<td>Lecture 4</td>
</tr>
<tr>
<td>Wed 24-Sep</td>
<td>15–17</td>
<td>LAB: 4</td>
<td>Practice: Control flows, Iterators</td>
</tr>
<tr>
<td>Week 5</td>
<td>Tue 30-Sep</td>
<td>12–14</td>
<td>Lecture 5</td>
</tr>
</tbody>
</table>
### Course Overview

#### Course Timetable

<table>
<thead>
<tr>
<th>Week 6</th>
<th>Tue 07-Oct</th>
<th>12–14</th>
<th>Lecture 6</th>
<th>Pandas for Data Analysis (Pandas)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Wed 08-Oct</td>
<td>15–17</td>
<td>LAB: 6</td>
<td>Practice: Pandas, Lab5 assigned</td>
</tr>
<tr>
<td>Week 7</td>
<td>Tue 14-Oct</td>
<td>12–14</td>
<td><strong>MIDTERM I</strong> (in class: 1 hour, Starts at 12)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Wed 15-Oct</td>
<td>15–17</td>
<td>No Lab</td>
<td></td>
</tr>
<tr>
<td>Week 8</td>
<td>Tue 21-Oct</td>
<td>12–14</td>
<td>Lecture 7</td>
<td>Geodetic Computations: pyproj, GeoPy, GeographicLib. Reading and Processing Shape files, Map Projections: Fiona, GDAL/OGR</td>
</tr>
<tr>
<td></td>
<td>Wed 22-Oct</td>
<td>15–17</td>
<td>LAB: 7</td>
<td>Practice: pyproj, GeoPy, GeographicLib, Fiona, GDAL/OGR</td>
</tr>
</tbody>
</table>

### Recommended/Reference Textbooks

Core GIS with Python

- **Week 6**
  - Lecture 6: Pandas for Data Analysis
  - Lab 6: Practice on Pandas, Lab 5 assigned

- **Week 7**
  - Lecture 7: Geodetic Computations
  - MIDTERM I

- **Week 8**
  - Lecture 8: Reading and Processing Shape files
  - Lab 7: Practice on pyproj, GeoPy, GeographicLib, Fiona, GDAL/OGR

### Map Visualization with Python

- **Week 9**
  - Lecture 8: Python 2D plotting library
  - Lab 8: Practice on Matplotlib

- **Week 10**
  - Lecture 9: Maps in Python
  - Lab 9: Practice on Basemap, Descartes

- **Week 11**
  - Lecture 10: Rendering Maps
  - Lab 10: Practice on Mapnik
## Processing Geospatial Data with Python

<table>
<thead>
<tr>
<th>Week 12</th>
<th>Tue 18-Nov</th>
<th>12–14</th>
<th><strong>Lecture 11</strong></th>
<th>Shapely</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Wed 19-Nov</td>
<td>15–17</td>
<td><strong>LAB: 11</strong></td>
<td>Practice: Shapely</td>
</tr>
<tr>
<td>Week 13</td>
<td>Tue 25-Nov</td>
<td>12–14</td>
<td><strong>Lecture 12</strong></td>
<td>Review/Catch up</td>
</tr>
<tr>
<td></td>
<td>Wed 26-Nov</td>
<td>15–17</td>
<td><strong>LAB: 12</strong></td>
<td>Practice: Shapely, Pandas, PySAL</td>
</tr>
<tr>
<td>Week 14</td>
<td>Tue 02-Dec</td>
<td>12–14</td>
<td><strong>Lecture 13</strong></td>
<td>Processing Geospatial Data with Python</td>
</tr>
<tr>
<td></td>
<td>Wed 03-Dec</td>
<td>15–17</td>
<td><strong>LAB: 13</strong></td>
<td>Course Project Due</td>
</tr>
<tr>
<td>Week 15</td>
<td>Tue 09-Dec</td>
<td>12–14</td>
<td></td>
<td>Review</td>
</tr>
<tr>
<td></td>
<td>Wed 10-Dec</td>
<td>15–17</td>
<td><strong>FINAL EXAM</strong></td>
<td>(in LAB)</td>
</tr>
</tbody>
</table>
Outline

Course Overview

Course Timetable

Assessment

Questions
Independent project

- Highlight your abilities and accomplishments in this course
- Come up with and implement a practical or fun program written by you in Python and focused on geospatial data analysis
  - This might end up being a group project but each person in the group should have a specific piece of code she/he is
Exams

- **Midterm exam (held in the lab):**
  - Most likely a set of python code implementations for a subset of smaller tasks demonstrating the proficiency in object oriented programming

- **Final exam (held in the lab):**
  - Most likely an overall exercise in program design and implementation with a specific focus on geospatial data analysis.
Grading

- Lab assignments (50%)
- Mid-term exam (15%)
- Independent project (15%)
- Final exam (20%)
Outline

Course Overview

Course Timetable

Assessment

Questions
Questions

- Questions
- Other issues
- You take the Blue or red pill?