GGS 590: Spatial Computing

Course Syllabus, Spring 2024, 3 Credits

Instructor: Dr. Edward Oughton

Email: eoughton@gmu.edu

Location: 2103 Exploratory Hall (in-person)

When: Monday 16:30 – 19:10

Learning Assistants (LAs): N/A

Pre-requisites: None.

Contact method: Blackboard discussion board for content related queries (preferred) and/or direct email for anything personal (eoughton@gmu.edu).

OVERVIEW & OBJECTIVES

It is imperative that spatial scientists have a comprehensive understanding of spatial computing methods and applications. Consequently, this class focuses on developing students with the basic to intermediate computing knowledge to generate programing applications within the Geographic Information Science (GIS) domain. We all engage with spatial computing applications on an almost hourly basis. For example, our smartphones utilize a wide range of spatial information to provide content, applications and services which we often take for granted, from suggested directions to Points of Interest (e.g., via Google Maps), through to location-targeted advertising. Within the scientific context, spatial computing can help us develop new place-based knowledge on topics related to the environment, society and the economy. Companies and governments utilize spatial computing to develop decision-support systems for asset management, natural resource assessment, labor management, and logistics tracking.

With this in mind, this class will introduce students to one of the most powerful and accessible programming languages – Python. Using both scripting and object-oriented programing approaches, students will learn how to initially reproduce the basic processing of points, linestrings and polygons often carried out elsewhere using Graphical User Interface GIS software such as ESRI ArcGIS Pro or QGIS. Before progressing to applying these powerful concepts to their own applications of choice. The focus is both on learning syntax and operations, but also developing critical thinking skills around how to apply spatial computing to problems of interest in the sciences. No programing experience is expected, meaning the class will progress from and introductory to intermediate programing stage. Advanced Python programmers will be given supplementary tasks to ensure sufficient learning.

LEARNING OUTCOMES

The key learning objectives of this class include:

1. Understanding the basics of computer programming (e.g., variables, functions, iteration etc.), as well as different design approaches (scripting versus object-oriented), with relevance to the spatial sciences.
2. Developing critical thinking skills for spatial computing uses, design approaches, and methodological choices.
3. Mastering important programming applications (e.g., Google Colab) and key spatial computing packages (e.g., shapely, geopandas etc.).

TIME COMMITMENT

While the class is designed to begin from scratch, assuming no existing programming experience, there will be a substantial time commitment. For example, when learning any new language (computer-based or natural), it takes significant effort to progress to fluency. Approximately 10-20 hours of weekly effort is expected outside of class (on assignments, exercises or further reading), with this effort hopefully rewarded later via the job market (as these skills usually translate to higher salaries). If you already have programing experience, this requirement is essentially minimized which may mean the first few weeks are relatively straightforward.

GGS COMPUTER LAB, ASSIGNMENTS, & EXPECTATIONS

GGS students have remote access to the GGS Virtual Computing Lab, although everything for this class should be available via the associated [GitHub page](#) (with content going up weekly). Affiliated Google Colab notebooks can be run via a web browser. Thus, all documents, notes and code are available from this single location.

If you do not intend to use the computer lab, then you will need a machine with at least 2 GB RAM, along with a fast and reliable broadband connection (e.g., > 10-20 Mbps). It may also be useful to have a web camera with a microphone in case any additional Zoom sessions are required.

Students will also need to have access to a general piece of GIS software to quickly check/inspect any spatial processing, e.g., via ESRI ArcGIS Pro (licensed) or QGIS (open-source and preferred). Please be aware this is a coding-focused class that will use either of these pieces of software as secondary tools. Students should not expect or seek to use these tools primarily in this class.

Assignments will be based on the lecture material you receive and will be administered via Blackboard. Generally, assignments will be set on a Monday and will be due the following Sunday evening (except when noted in the Course Schedule). Late work will be penalized 20% for each day late. Late submissions will only go unpenalized for documented medical reasons or by previous agreement with the instructor (e.g., raised at the time the assignment is set).

Each student gets the opportunity to drop the two worst performing assignments from the overall score at the end of the semester. Take comfort in the fact that it is highly unlikely that three events justifying

extenuating circumstances would occur in a single semester. Thus, the course grading criteria is accounting for unfortunate events. No additional requests will be accommodated unless the extenuating circumstances are highly serious (in which case the university and course director will already be aware of the issue).

The overall grade is comprised of three key sets:

GRADING		
Assessment	**Points**	**%** (of final grade)
Tests/Assignments (6)	100 each	50%
Coursework project	100	25%
Exam	100	25%

The exam will consist of 20 multiple choice questions, on the basics of spatial computing.

Grading will be based on the following cutoff values, although the instructor reserves the right to alter the values at the end of the course:

A (93%), A- (90%), B+ (87%), B (83%), B- (80%), C+ (77%), C (73%), C- (70%), D (60%)

The coursework project will include the use of techniques taught throughout the entire semester but applied to your own research topic. Students will be expected to submit assignments online through Blackboard. Only Adobe PDF (.pdf) file formats will be accepted (students can save documents to a .pdf format for submission).

OPTIONAL TEXTS

There are many open and free resources for learning Python. Importantly, all the information you require will be provided in the course. However, should you require additional information, for example relating to programming languages, there are many options. Check out Al Sweigart's [Automate the Boring Stuff with Python] which is free to read and provides practical programming for total beginners.

OFFICE HOURS AND INSTRUCTOR INTERACTION

Office hours will be held on Mondays from 16.00 – 16.30 hrs prior to each class. Each student is expected to attend office hours at least once per semester. When emailing, a timely response is expected during office hours Monday-Friday. If you have a course-related question, the first port of call will be to place it on the Blackboard discussion board. This is because other people may ask similar questions, so this becomes a shared knowledge base everyone can access. If you need to speak about something more personal with the instructor, then you can reach out via email. Drop-in virtual Q&A sessions can also be held by request (e.g., Wednesday 8pm) to seek advice on assignments and coursework.

PROBLEM SOLVING

It is inevitable that problems will arise, especially when working with Python code. Therefore, it is essential that students follow a set of key procedures when dealing with any coding issues encountered. These are as follows:

1. Copy and paste any error messages into a search engine (e.g., Google). Someone else will already have had the same problem, so investigate how other researchers solved similar issues.

2. Explore Stack Overflow questions and answers. When other programmers have been stuck, they post them publicly on Stack Overflow asking for answers, making this a great resource.

3. If you still cannot solve the problem, either post a public question on Stack Overflow, or follow the instructions for posting a public question on the GGS366 Blackboard discussion board.

ACADEMIC INTEGRITY

GMU has an Honor Code with clear guidelines regarding academic integrity. Three fundamental and rather simple principles to always follow are that: (1) all work submitted be your own; (2) when using the work or ideas of others, including fellow students, give full credit through accurate citations; and (3) if you are uncertain about the ground rules on a particular assignment, ask for clarification. No grade is important enough to justify academic misconduct (e.g., plagiarism). Another aspect of academic integrity is the free play of ideas. Vigorous discussion and free speech debate are encouraged, with the expectation that all aspects of the class will be conducted with civility and tolerance for differing ideas, perspectives, and traditions.

GMU EMAIL ACCOUNTS

Students must use their MasonLive email account to receive important University information, including messages related to this class. See http://masonlive.gmu.edu for more information. Please do not email the instructor from a non-GMU email account.

OFFICE OF DISABILITY SERVICES

If you are a student with a disability and you need academic accommodations, please contact the instructor and the Office of Disability Services (ODS) at 993-2474, http://ods.gmu.edu. All academic accommodations must be arranged through the ODS.

GMU RESOURCES

The Writing Center: https://writingcenter.gmu.edu

University Libraries, Ask a Librarian: https://library.gmu.edu/ask

Counseling and Psychological Services: https://caps.gmu.edu

University Catalog: https://catalog.gmu.edu

University Policies: https://universitypolicy.gmu.edu

COURSE OUTLINE

Week	Topic	Coursework
Week 1: **Aug 26th**	Introduction and course overview. Intro to Python and the Google Colab environment.	Background reading
Week 2: **Sept 2nd**	Labor Day	-
Week 3: **Sept 9th**	Variables, data types, and functions.	Assignment 1
Week 4: **Sept 16th**	Data structures.	Assignment 2
Week 5: **Sept 23rd**	Loops, loop control logic and Shapely.	Assignment 3
Week 6: **Sept 30th**	Intro to Pandas and Geopandas.	-
Week 7: **Oct 7th**	Scripting (part 1). Spatial processing using Geopandas.	Assignment 4
Fall Break (Oct 14th)		
Week 8: **Oct 21st**	Scripting (part 2). Spatial processing using Geopandas.	Assignment 5
Week 9: **Oct 28th**	Intro to object-oriented programming (part 1).	-
Week 10: **Nov 4th**	Intro to object-oriented programming (part 2).	Assignment 6
Week 11: **Nov 11th**	Shortest-path routing (Kruskal's Algorithm)	Coursework Project
Week 12: **Nov 18th**	Coursework Project Introduction	Coursework Project
Week 13: **Nov 25th**	Coursework Project	Coursework Project
Week 14: **Dec 2nd**	Coursework Project	Coursework Project
Week 15: **Dec 9th**	Coursework Project Submission / Exam	-

Note: The course schedule is tentative and is subject to revision by the instructor