GEOG 5222: GIS Programming and Algorithms

The Ohio State University
Autumn 2023

Course URL: http://carmen.osu.edu
Time: Monday and Wednesday, 9:35 - 10:55 AM
Location: Derby Hall 135

Instructors

Prof. Ningchuan Xiao | xiao.37@osu.edu

Ms. Ruixuan Ding | ding.796@buckeyemail.osu.edu

Office hours

Day	Room	Time and zoom link	Instructor
Monday	1160 Derby Hall	11 AM - 12 PM	Dr. Xiao
Monday	1170 Derby Hall	2:30 - 3:30 PM	Ms. Ding
Tuesday	1170 Derby Hall	11:30 AM - 12:30 PM	Ms. Ding
Wednesday	1160 Derby Hall	11 AM - 12 PM	Dr. Xiao

This class is about coding.

More specifically, we learn how to program a computer to handle spatial data in this class. It is obvious, one may argue, that today's software are often quite easy to use and they are also powerful. So why should we do coding? In his book *The Mythical Man-month*, Frederick P. Brooks, Jr. talked about the reward of coding as the sheer joy of making things. Others agreed. In a 1984 New York Times piece, Erik Sandberg-Diment gave three reasons of programming: "First, it allows you to develop software that is not available commercially, and in some cases it lets you customize purchased software to serve your specific needs better. Second, programming can be fun. If you enjoy working on puzzles, programming may well turn out to be more pleasurable than solving The Times crossword puzzle or Dr. Crypton's mind-bending puzzle page in Science Digest. Third, there is the intellectual exercise, the honing of logic skills and learning to learn, stressed by pedagogues as a perfect reason to have computers

available in schools for pupils from kindergarten age on up." In a blog at Invent with Python, Al Sweigart pointed out that "Programming transforms your computer from a home appliance to a power tool." It is apparent that writing computer code is becoming an essential part of life. As Steve Jobs once put it, "everybody in this country should learn to program a computer...because it teaches you how to think." So in one word: coding is fun.

In this class, we aim to help students gain freedom in dealing with spatial data though programming. After completing this class, students will develop understanding on how spatial data works and how to write their own code to handle the data, instead of relying on existing (especially commercial) software packages. We use the powerful and very popular (see PYPL or TIOBE) Python programming language in this class. We start from the basics of Python. About half way through the semester, we should have developed sufficient knowledge about programming with Python and will start to use it on spatial datasets. Most of the materials do not rely on existing software packages, which will provide an intuitive way of understanding spatial data and spatial data processing. By the end of the semester, students should understand the fundamental considerations of computational issues in spatial data processing and should be able to write Python code to complete various tasks of using spatial data.

Course learning outcomes

By the end of this course, students should be able to:

- Understand fundamental constructs and styles of programming using Python
- Write Python programs to conduct basic data processing and visualization
- Develop and test Python programs to process and visualize spatial data
- Develop and test Python programs for basic geometric processing
- Understand basic spatial indexing methods for point data sets
- Understand the basics of computational issues such as time complexity and indexing

Textbooks and Other Materials

Two textbooks are required for this class:

- Allen B. Downey. 2016. **Think Python**. 2nd Edition. Sebastopol, CA: O'Reilly Media.
- Ningchuan Xiao. 2016. **GIS Algorithms**. London, UK: SAGE Publications.

A GitHub site repository at http://github.com/gisalgs will be used intensively for most part of the class. We will only access this repository through a web browser. This is a public website and everyone can access. The detailed accessibility statement of GitHub can be found at https://government.github.com/accessibility. The privacy statement of GitHub can be found at https://help.github.com/en/github/site-policy/github-privacy-statement.

Tutorials and lecture notes will be made available through the course schedule page on Carmen. There are also numerous useful online sources for learning Python. Al Sweigart maintains a web site for his book at https://automatetheboringstuff.com. Also, The Hitchhiker's Guide to Python! (https://docs.python-guide.org) has information that can be extremely practical for many beginners. The official Python Tutorial (https://docs.python.org/3/tutorial) is a good place to find the details on most of the topics in Python coding (make sure to choose the right version on top of the page).

Schedule

The overall course contents are divided into a few major topics:

Weeks 1-8: Python programming language
Weeks 9-10: Geometric algorithms
Weeks 11-15: Spatial indexing
Weeks 15-16: Applications

A detailed schedule can be found on the course page on Carmen. Students should check this page frequently as it will be updated whenever new materials are made available. Important notes regarding the course will also be posted on the home page.

Course organization and assignment information

Course contents are organized into modules. A typical module is designed for 5 instructional days (weekends and holidays are not included), includes two meets, and will not end on a meet day. Not all the meet days are used for the modules, as some will be for quizzes and other tasks. See the detailed schedule on Carmen for the specific tasks.

- **Exercises:** There are some relatively small questions associated with each module. Exercises are due by the last instructional day in a module.

- **Pop quizzes:** We will have a number of *unannounced* pop quizzes throughout the semester. Each of these quizzes will include questions related to recent topics

covered in the class.

- **Homework:** For each major topic on the course schedule, there will be a homework assignment with more in-depth questions. These questions typically require more intensive coding efforts and students have about two weeks to finish.

- **Quizzes:** Quiz will be given after we finish a major topic. These are open-book quizzes that will require students use the computer to answer. These quizzes are intensive coding exercises that require students to finish in 80 minutes.

- **Before you go...quizzes:** There is an online and open-book quiz at the end of a module. The questions are related to the topics in the current module, but may also be cumulative. This quiz is timed. Each student has three attempts and the highest score will be used.

- **Final projects:** A list of guided questions will be provided in the middle of the semester and each student will choose to answer one of these questions. The questions will be based on the topics covered in the semester. Only sufficient information will be provided, which means I will not provide all the implementation details for the questions and students must find solutions by themselves. There will also be open questions that give students a lot of freedom to choose their own project. The open questions will require a short proposal from the students.

Grading

Assignment category	Percent
Exercises	25
Homework	20
Pop quizzes	5
Before you go...quizzes	10
Quizzes	20
Final projects	20
Total	100

Faculty feedback and response time

The following list outlines the intended availability of the instruction team throughout the course. (Remember that you can call 614-688-HELP at any time if you have a technical problem.)

Grading and feedback For the exercises, students should be able to see the grade and feedback in a week after the due dates. For large assignments (homework and projects), the timeframe to expect feedback is generally 1 to 2 weeks.

E-mail E-mails will be replied within 48 hours, and we aim for 24 hours during school days.

Office Hours In addition to the virtual office hours, students may request other times to meet with the instruction team. Please give at least 24 hours notification and it will only be available during school days.

Important Class Policies

Do your own work. Collaboration is healthy and often necessary, but each student should definitely finish the work individually. Please see below for more information about academic misconduct.

Do your OWN work. This policy also applies to the use of generative AI such as ChaptGPT. We will treat these as tools that can help us learn, but they are not going to replace us writing the code. There will be numerous situations in this class when we need to explain why the code works, which will only come from a natural understanding of programming.

Due date and make-ups. Due dates for all assignments will be clearly stated on Carmen. If you need any extensions, you must ask for it **prior to** the due date. Extensions will not be granted if requested after the due date. This applies to all assignments. For a quiz, the due date is the start of the class when the quiz is scheduled. Make-up quiz will only be given to those who request before the quiz starts.

Late assignments. Late submissions may be accepted up to a week past the due date. Each day after the due date will incur a 10% penalty. The total penalty will be up to 50%. Five to seven days late will only receive 50% credit of the grade you would have received if it was submitted on time. If you contact me ahead of time for deadline adjustments you will not incur any penalty. Please note this may not apply to every assignment. The final project, for example, has a firm deadline that cannot be changed. Please refer to Carmen for due dates.

Deliverables. All deliverables must be submitted on Carmen as specified in the homework/project instructions. There will be absolutely **no email submissions**. Email submissions of work for this class will not be acknowledged and will not be accepted.

Communication. The only official way to communicate with the TA and myself is through our OSU email addresses as listed on the top of the syllabus. We cannot guarantee that we will reply messages through any other methods. We normally will reply emails in a instructional day.

Student with Disabilities

The University strives to make all learning experiences as accessible as possible. If you anticipate or experience academic barriers based on your disability (including mental health, chronic or temporary medical conditions), please let me know immediately so that we can privately discuss options. You are also welcome to register with Student Life Disability Services to establish reasonable accommodations. After registration, make arrangements with me as soon as possible to discuss your accommodations so that they may be implemented in a timely fashion. The web site of SLDS is slds.osu.edu, and its contact information is slds@osu.edu, 614-292-3307, 098 Baker Hall, 113 W. 12th Avenue.

Weather or other short-term closing

Should in-person classes be canceled, I will notify you as to which alternative methods of teaching will be offered to ensure continuity of instruction for this class. Communication will be via email, and in-class or Carmen announcements.

Policy on plagiarism and academic misconduct

If I suspect that a student has committed academic misconduct in this course, I am obligated by University Rules to report my suspicions to the Committee on Academic Misconduct. The term "academic misconduct" includes all forms of student academic misconduct wherever committed, illustrated by, but not limited to, cases of plagiarism and dishonest practices in connection with examinations. It is the responsibility of the Committee on Academic Misconduct to investigate or establish procedures for the investigation of all reported cases of student academic misconduct. Instructors shall report all instances of alleged academic misconduct to the committee (Faculty Rule 3335-5-487). For additional information, see the Code of Student Conduct at http://studentlife.osu.edu/csc/.

The following is a list of other useful sources of information on academic misconduct and academic integrity:

- The Committee on Academic Misconduct web pages: COAM Home
- Ten Suggestions for Preserving Academic Integrity
- Eight Cardinal Rules of Academic Integrity
- How to Avoid Plagiarism

Religious accommodations

It is Ohio State's policy to reasonably accommodate the sincerely held religious beliefs and practices of all students. The policy permits a student to be absent for up to three days each academic semester for reasons of faith or religious or spiritual belief.

Students planning to use religious beliefs or practices accommodations for course requirements must inform the instructor in writing no later than 14 days after the course begins. The instructor is then responsible for scheduling an alternative time and date for the course requirement, which may be before or after the original time and date of the course requirement. These alternative accommodations will remain confidential. It is the student's responsibility to ensure that all course assignments are completed.