Oregon State University

Course Name: Programming for Geospatial Analysis

Course Number: GEOG 462/562

Course Credits: 4

Class Meeting Times

Lecture MWF 9 – 9:50 AM
Lab: M 2 – 3:50 PM

Prerequisite and/or Corequisite

(GEOG 361 [C-] or GEOG 362 [C-])
GEOG 561

Catalog Course Description

Explores the extension of geographic information systems (GIS) through programming. No prior programming experience is expected. Examines design and writing of computer programs to automate geospatial analysis. Apply basic principles of programming languages to write, apply, and interpret results of scripts in software such as Python. Develop custom functions and analyze coding information to learn new skills. Emphasizes building foundations and pragmatic approaches to designing and constructing programs and scripts for geospatial analysis.

Instructors:

Lectures: Robert Kennedy, robert.kennedy@oregonstate.edu
Strand Ag 334b / (541) 737 6332
Office hours: Tuesday and Friday 2-3pm, or by appointment

Labs: Ashraful Islam (islamkm@oregonstate.edu)
Office hours: TBD

Course components

In every module, you will watch online lecture videos where I introduce content. You will then weigh in on a discussion Canvas about them, and we'll discuss the issues that arise in person. You'll also do a homework each module, and every approximately two weeks take an in-person quiz. These will help you with the conceptual foundations.

You will pair that with practical implementation in lab exercises every module. Additionally, you'll work on a project in the second half of the term that allows you explore topics of interest to you through coding.

Grad students will additionally explore how Python or coding in general interact with their field of interest, and present their findings in oral presentations to the rest of the class.

Readings

I do not require a textbook. However, these texts are considered useful but not required:
"Programming ArcGIS Pro with Python" Eric Pimpler. Publisher: GeoSpatial Training Services. Available via Amazon.
"Learning Python" Most recent edition, by Mark Lutz. O'Reilly publishing
I also recommend keeping up with the online readings in the tutorial on Python.org. I'll post these in a separate "Readings" document through Canvas.

Modules

We will have nine modules through the term. Each module runs Thursday through Wednesday, with some possible exceptions to accommodate special scheduling issues.

- Thursday: I will post videos of the content for the module. I will also post a homework exercise.
- Friday through Sunday: In lieu of coming to class on Friday morning, you will watch the videos and provide structured responses on Canvas
- Monday: We will meet in class to dig deeper into the issues that arise in the discussion. Monday afternoons, you'll also attend lab in the Digital Earth Lab (Wilkinson 210)
- Wednesday morning you'll turn in the homework exercise and come to class, where we will discuss that exercise. Roughly every other week, you'll take a short in-person quiz on that material.

Labs

Programming requires computers! Labs will be critical. PLEASE attend lab! The lab is where the rubber hits the road, and it's the best time to get help from your lab teaching assistant.

We will have lab session in Wilkinson 210 on Mondays from 2-4pm. Your TA will introduce the basic material and then help you as you work through it. You'll be using software that is on the lab machines. You can also install the software on your own computer if you have a PC, and some of it if you have a Mac, but for practical reasons we are not able to help with system debugging on installation. The lab room will be open throughout the day during weekdays, but not available on weekends.

As you work through each lab's exercise, you're going to be writing code into Python files – these files will have the filename extension ".py", or, when we start out with Notebooks, ".ipynb" Additionally, you'll be keeping a journal of your work (details in the lab document assignments).

We use an immediate response approach to get you get you feedback on your coding. In a standard model, one might turn in all your lab materials and the TA would check your work and give you feedback, but this could take up to a week – when your brain has already been filled with the material of the next week's lab! Thus, our model is to have you turn in your files on a Canvas assignment, and then you will *immediately* be given access to a coding *answer key* where we provide examples of ways to handle the lab exercises – giving you immediate feedback while your logic is still fresh in your mind.

You will review the answer key and write a short reflection on how your structure, logic, and syntax agreed and disagreed with what provided. Details will be available in Canvas. Additionally, you'll turn in examples of how you have used commenting in your code, and the file you used for journaling.

The TA will go through your reflections and make sure you're getting the learning objectives for the lab and that you've done sufficient commenting and journaling. This gives the TA more time to work with folks individually on the challenges, rather than spending time reviewing code that folks already understood.

All of this must be done by the beginning of the next week's lab.

Lab discussions

Our goal is to get everyone to learn the material, not to arbitrarily weed people out. Programming can be frustrating because even the smallest error stops the entire process in its tracks. Odds are that any small hurdle (or large roadblock) you solve will be experienced by one of your classmates, and we can move forward as a class better if learn from each other. We want to use a Canvas discussion to ease the process of building collective knowledge in the class. In each week's module on Canvas, I will set up a new discussion focused around the week's lab. Post ideas, revelations, questions, problems, etc. We hope to make these a place where you can learn from each other – the learning in the labs is not a zero-sum game, so the more we help each other, the better everyone will do.

Lab grading

The specifics vary by lab, but generally include:

- Points for turning in the initial file(s)
- Points for your reflection
- Points for appropriate commenting and journaling.

Homeworks

Homeworks will be focused on the concepts of the week, and will be turned in on Canvas where you can immediately see our answers. You are welcome to use A.I. to help you learn, but note that you must retain the information to be able to take the quizzes.

Quizzes

Programming is less about memorization and more about clear thinking. Increasingly, A.I. is making this even more important: our role is becoming one of managing the A.I. tools into an overarching structure.

That said, to adequately utilize A.I. to code, you need to build a mental model yourself of how the computer thinks, and how to encourage the A.I. to structure code appropriately. The quizzes give you a chance to test your knowledge of those basic concepts.

Projects

Projects allow you to specialize your learning of Python, and to solve a challenge greater than what we can achieve in a single lab. As with Labs, we will conduct projects in groups of three, and each group must contain at least one grad and one undergrad member.

The project will progress by week as follows:

Module 3: The pitch

We will aim to have only 6-8 projects. Because there are more students than projects, we want to gravitate to those with the most interest. Thus, in module 3 of the term, **all grad students** must pitch an idea for a project to the class on Canvas. Undergrads may make a pitch, but need not. The pitch will be done in writing on a discussion on the Canvas site, and others in class will use comments to indicate which ones they would be interested in doing.

Themes could include: automation of a repeated task, exploration of geospatial programming using open-source rather than Arc tools, development of new functionality, mapmaking automation, development of an Arc Toolbox, or something you dream up.

Module 4: Coalescence
You'll read through the project pitches and indicate which ones sound cool, and I'll do my best to group people into groups that make sense by the end of the week.

Module 5: Group Meeting and Planning
In module 5, you'll arrange amongst yourselves to meet with your group. The goal is to assign roles and begin developing project plans, and articulate these into a pseudocode document and workplan with individual roles spelled out and target effort scores.

Module 6: Pseudocode and workplan / meet with Kennedy
In module 6, you'll turn in your work plan. I'll start meeting with each group for a half hour to discuss the plan and the group.

Module 7: Data for project due/ meet with Kennedy
Turn in a written description of data that the group has acquired for the project. All data must be **in hand**. It is not sufficient to simply know the whereabouts of the data but not have it in hand. Consider this requirement when designing the project.

Modules 8 and 9: Focus on projects.

Finals week: All groups will present their project during a live symposium during the final-exam block in finals week. Final deliverables will be the presentation and a two-page double-spaced summary of the work, and the Python code.

Grading of project (40 points for undergrads, 50 points for grads)
Each group member receives the same score for these components:

5 points:	Pseudocode and work plan
5 points:	Data writeup
10 points:	Code
10 points:	Presentation and 2-page writeup

Each group member receives an individual score for these components:

10 points:	Project pitch (grad students only)
10 points:	Self-assigned effort score

Graduate student reports

Working in pairs (or groups of three, as necessary), graduate students will report on the use of Python (or any other geospatial programming language) within a specific discipline (their own, or of their group-mate). This can blend discipline-specific Python libraries, web pages or projects, or papers. The following pieces should be included:

- What problem is this discipline trying to solve using geospatial programming?
- What new ways of looking at the problem does programming enable?
- What are specific examples of how people have used it?
- Develop a pseudocode example of how you might use it to solve an actual problem with a real dataset in your own discipline.

The report will consist of a one-page double-spaced (250 words max) written report, and 5-minute presentation + 5 minute discussion with the class in weeks 8-10 of the class.

Grading

GEOG462:
45% labs
10% homeworks
15% quizzes
20% project
10% participation in class: participate in discussions and class activities

GEOG562
30% labs
10% homeworks
15% quizzes
25% project
10% grad report
10% leadership and engagement: leadership in class discussions

Final grades will be based on standard percentile breakdowns on a non-stretched scale: 90 to 100% of points will earn you an "A", 80-90% of points a "B" etc. (with + and – at three-percent breaks – e.g. 90-93 = A-, 87-89.99 = B+, etc.).

Course-specific Learning outcomes

By the end of GEOG 462/562, students will be able to:

1. Apply the programming and scripting language Python at the console level.
2. Describe and explain Python scripts written by others
3. Recognize and apply basic principles of programming languages, including variable types, conditionals and control flow, functional programming, and object-based programming
4. Predict expected behavior in a Python script
5. Employ an integrated development environment to run and debug Python scripts
6. Write stand-alone Python scripts that are robust to exceptions
7. Apply ArcGIS's ArcPy module to automate spatial analysis from the command line and from within scripts
8. Read, manipulate, and write both vector and raster data from scripts
9. Develop custom Python functions and classes
10. Analyze Internet-based coding information to learn new skills.
11. Design a geospatial coding project, acquire data, carry out analysis and report results in a student symposium

Additional learning outcome for graduate students

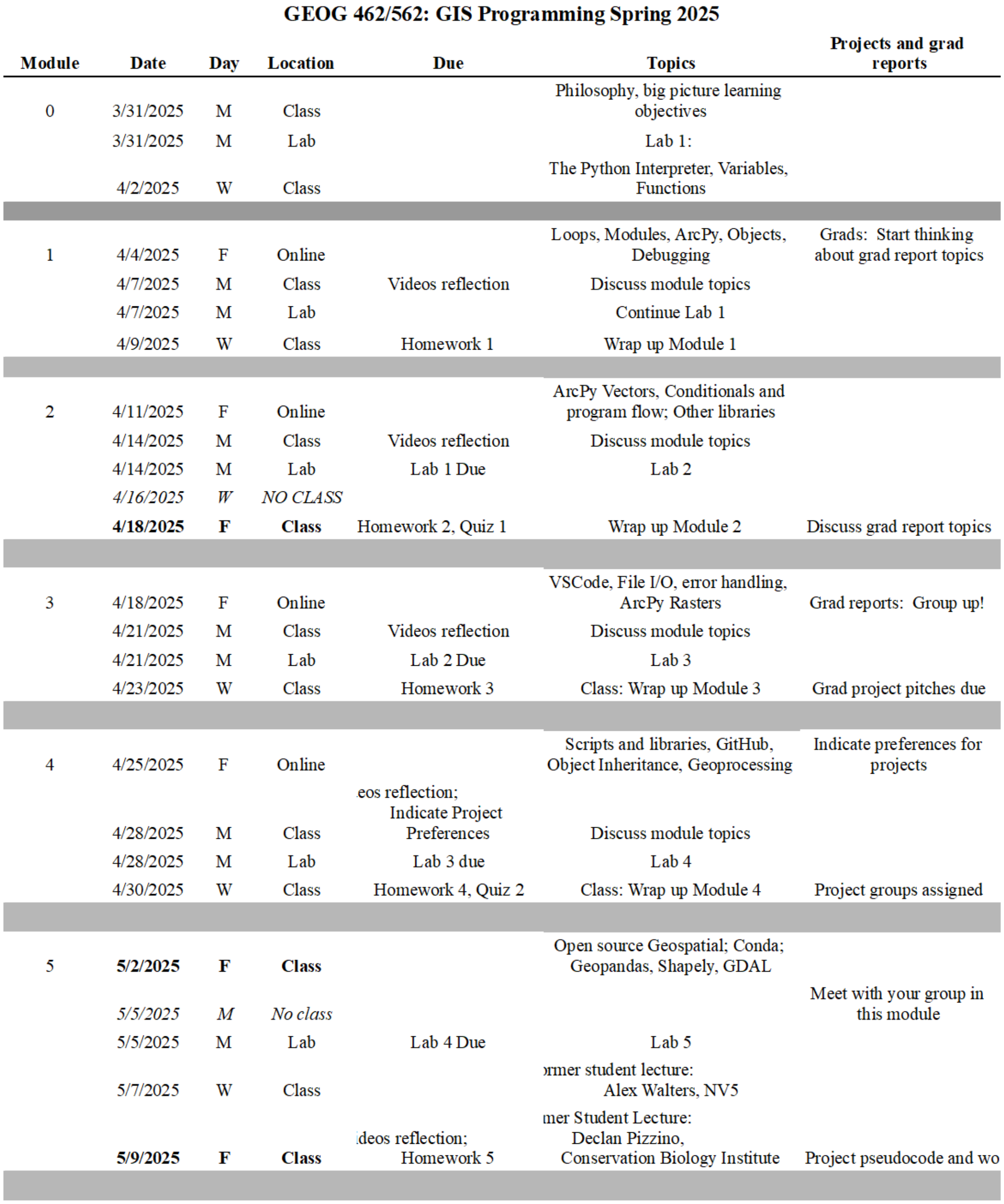
1. Describe and summarize how programming is used in a discipline of interest
2. Integrate a discipline-specific approach to programming with concepts learned in this course and report in written and spoken form

Distinction between 562 and 462

To distinguish 562 from 462, graduate students have the additional responsibilities:

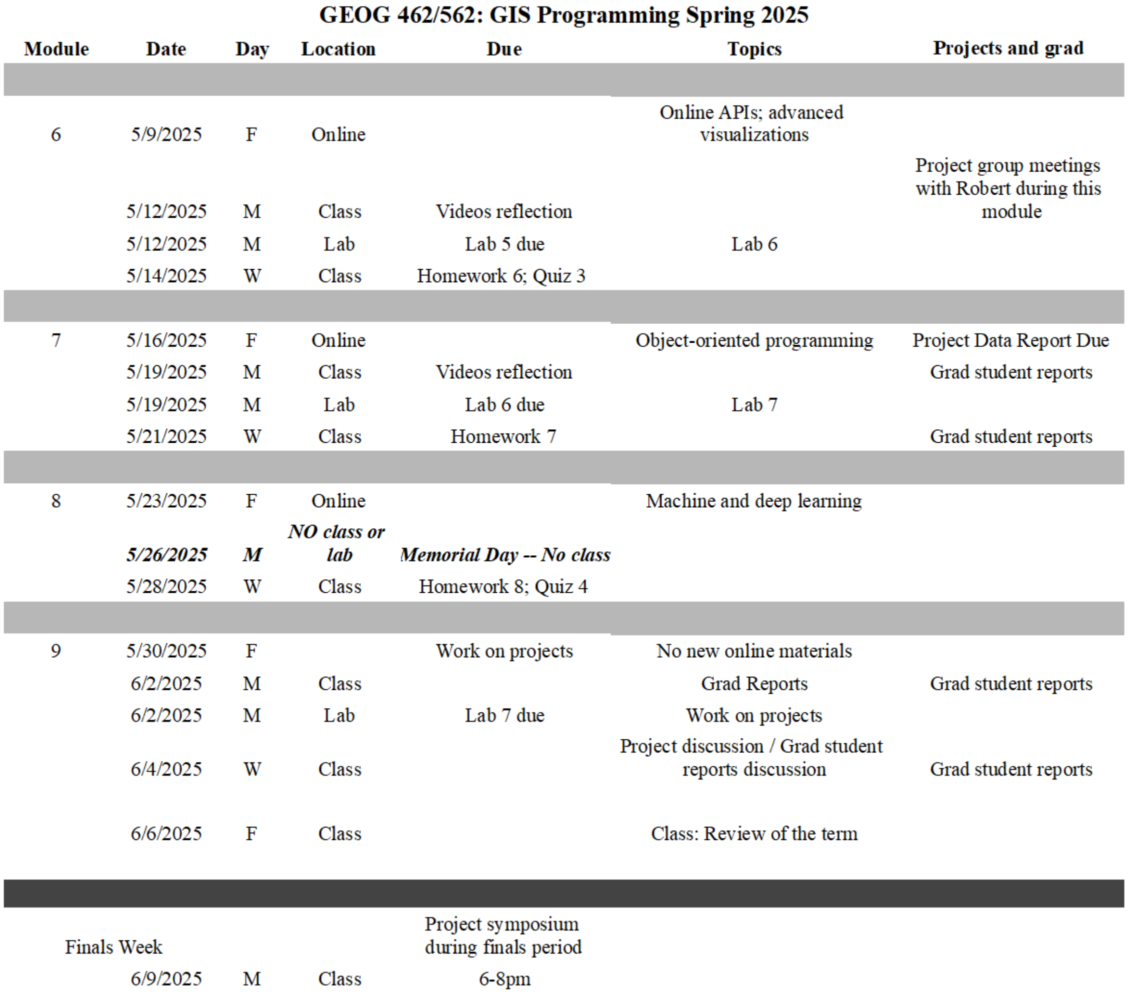
1. Graduate student reports
2. Expected leadership role in project groups
3. Generation of project ideas
4. Expected leadership during class discussions.

Calendar



GEOG 462/562: GIS Programming Spring 2025

Module	Date	Day	Location	Due	Topics	Projects and grad reports
0	3/31/2025	M	Class		Philosophy, big picture learning objectives	
	3/31/2025	M	Lab		Lab 1:	
	4/2/2025	W	Class		The Python Interpreter, Variables, Functions	
1	4/4/2025	F	Online		Loops, Modules, ArcPy, Objects, Debugging	Grads: Start thinking about grad report topics
	4/7/2025	M	Class	Videos reflection	Discuss module topics	
	4/7/2025	M	Lab		Continue Lab 1	
	4/9/2025	W	Class	Homework 1	Wrap up Module 1	
2	4/11/2025	F	Online		ArcPy Vectors, Conditionals and program flow; Other libraries	
	4/14/2025	M	Class	Videos reflection	Discuss module topics	
	4/14/2025	M	Lab	Lab 1 Due	Lab 2	
	4/16/2025	*W*	*NO CLASS*			
	4/18/2025	**F**	**Class**	Homework 2, Quiz 1	Wrap up Module 2	Discuss grad report topics
3	4/18/2025	F	Online		VSCode, File I/O, error handling, ArcPy Rasters	Grad reports: Group up!
	4/21/2025	M	Class	Videos reflection	Discuss module topics	
	4/21/2025	M	Lab	Lab 2 Due	Lab 3	
	4/23/2025	W	Class	Homework 3	Class: Wrap up Module 3	Grad project pitches due
4	4/25/2025	F	Online		Scripts and libraries, GitHub, Object Inheritance, Geoprocessing	Indicate preferences for projects
	4/28/2025	M	Class	eos reflection; Indicate Project Preferences	Discuss module topics	
	4/28/2025	M	Lab	Lab 3 due	Lab 4	
	4/30/2025	W	Class	Homework 4, Quiz 2	Class: Wrap up Module 4	Project groups assigned
5	**5/2/2025**	**F**	**Class**		Open source Geospatial; Conda; Geopandas, Shapely, GDAL	
	5/5/2025	*M*	*No class*			Meet with your group in this module
	5/5/2025	M	Lab	Lab 4 Due	Lab 5	
	5/7/2025	W	Class		rmer student lecture: Alex Walters, NV5	
	5/9/2025	**F**	**Class**	ideos reflection; Homework 5	mer Student Lecture: Declan Pizzino, Conservation Biology Institute	Project pseudocode and wo



GEOG 462/562: GIS Programming Spring 2025

Module	Date	Day	Location	Due	Topics	Projects and grad
6	5/9/2025	F	Online		Online APIs; advanced visualizations	
	5/12/2025	M	Class	Videos reflection		Project group meetings with Robert during this module
	5/12/2025	M	Lab	Lab 5 due	Lab 6	
	5/14/2025	W	Class	Homework 6; Quiz 3		
7	5/16/2025	F	Online		Object-oriented programming	Project Data Report Due
	5/19/2025	M	Class	Videos reflection		Grad student reports
	5/19/2025	M	Lab	Lab 6 due	Lab 7	
	5/21/2025	W	Class	Homework 7		Grad student reports
8	5/23/2025	F	Online		Machine and deep learning	
	5/26/2025	***M***	***NO class or lab***	***Memorial Day -- No class***		
	5/28/2025	W	Class	Homework 8; Quiz 4		
9	5/30/2025	F		Work on projects	No new online materials	
	6/2/2025	M	Class		Grad Reports	Grad student reports
	6/2/2025	M	Lab	Lab 7 due	Work on projects	
	6/4/2025	W	Class		Project discussion / Grad student reports discussion	Grad student reports
	6/6/2025	F	Class		Class: Review of the term	
Finals Week	6/9/2025	M	Class	Project symposium during finals period 6-8pm		

Course Policies

Late stuff

Except in unusual circumstances, if you miss an event – a class, a quiz, a due date – without checking with us in advance, you lose that event. Check in advance if you have a problem meeting a due-date for a lab, presentation, or a quiz. If you're sick but not deathly ill, try to at least email me. I'm a reasonable guy.

Please don't turn stuff in late – we need to build on each week's material. You'll lose 10% of points for the first day late, 20% for each day thereafter. In other words, from 0 to 24 hours late, the maximum score is 90%; for 24 to 48 hours late, the max is 70%; for 48-72 hours late the max is 50%, etc.

The one exception: I'll give each person a freebie to turn in one assignment up to two days late without penalty – except the final research proposal, which has to be in on time so I can get your grade calculated. If you're using your freebie, mark it clearly IN THE TEXT OF THE DOCUMENT! Thanks.

Email etiquette

- I'll try to answer all appropriate emails within 24 hours. Don't expect an instant response, though.
- Include "GEOG4/562" in the subject line of all emails with me so I can see it within the sea of other emails.
- I'll respond quicker to emails that have clear subject lines.
  - A good subject line: "GEOG4/562: Problem saving file to my data directory"
  - Bad subject lines: "question….", "today", "confused", etc.
- I'll respond quicker to emails where the core question in the body of the email is put in bold face, and where separate ideas are in separate paragraphs. Stream-of-consciousness text works fine for some fiction, but I find it hard to parse.
- Please use full sentences in your email and make sure you've thought through the email. I reserve the right to not respond to lazy emails…

Statement on use of Artificial Intelligence (A.I.)

Coding has been transformed by A.I. Increasingly, programmers manage code chunks written by A.I. tools. This does not replace the need to understand how the computer thinks, how to organize large workflows, how to test whether it is providing the right logic, and how to effectively leverage coding to ask scientific questions. Thus, we encourage the appropriate use of A.I. in this course. It can be used to help learn, but not in place of learning.

Academic Calendar

All students are subject to the registration and refund deadlines as stated in the Academic Calendar: https://registrar.oregonstate.edu/osu-academic-calendar

Statement Regarding Students with Disabilities

Accommodations for students with disabilities are determined and approved by Disability Access Services (DAS). If you, as a student, believe you are eligible for accommodations but have not obtained approval please contact DAS immediately at 541-737-4098 or at http://ds.oregonstate.edu. DAS notifies students and faculty members of approved academic accommodations and coordinates implementation of those accommodations. While not required, students and faculty members are encouraged to discuss details of the implementation of individual accommodations."

Student Conduct Expectations link:

https://beav.es/codeofconduct

Student Bill of Rights

OSU has twelve established student rights. They include due process in all university disciplinary processes, an equal opportunity to learn, and grading in accordance with the course syllabus: https://asosu.oregonstate.edu/advocacy/rights.

Reach Out for Success

University students encounter setbacks from time to time. If you encounter difficulties and need assistance, it's important to reach out. Consider discussing the situation with an instructor or academic advisor. Learn about resources that assist with wellness and academic success at oregonstate.edu/ReachOut. If you are in immediate crisis, please contact the Crisis Text Line by texting OREGON to 741-741 or call the National Suicide Prevention Lifeline at 1-800-273-TALK (8255)

Student Evaluation of Courses

During Fall, Winter, and Spring term, the online Student Evaluation of Teaching system opens to students the Wednesday of week 8 and closes the Sunday before Finals Week. Students will receive notification, instructions and the link through their ONID email. They may also log into the system via Online Services. Course evaluation results are extremely important and used to help improve courses and the learning experience of future students. Responses are anonymous (unless a student chooses to "sign" their comments, agreeing to relinquish anonymity) and unavailable to instructors until after grades have been posted. The results of scaled questions and signed comments go to both the instructor and their unit head/supervisor. Anonymous (unsigned) comments go to the instructor only.