Course Change Request

New Course Proposal

Date Submitted: 02/10/21 1:09 pm

Viewing: GGS 366 : Spatial Computing

Last edit: 02/10/21 1:09 pm

Changes proposed by: tleslie

Are you completing this form on someone else's behalf?

In Workflow

1. **GGS Chair**
2. **SC Curriculum Committee**
3. SC Associate Dean
4. Assoc Provost-Undergraduate
5. Registrar-Courses
6. Banner

Approval Path

1. 02/02/21 4:05 pm Nathan Burtch (nburtch): Rollback to Initiator
2. 02/11/21 4:13 pm Nathan Burtch (nburtch): Approved for GGS Chair

Yes

Requestor:

Name	Extension	Email
Andreas Zuefle	3-1210	azufle@gmu.edu

Effective Term: Fall 2021

Subject Code: GGS - Geography & Geoinformation Science **Course Number:** 366

Bundled Courses:

Is this course replacing another course? No

Equivalent Courses: GGS 210 - Introduction to Spatial Computing

Catalog Title: Spatial Computing

Banner Title: Spatial Computing

Will section titles vary by semester? No

Credits: 3

Schedule Type: Lecture

Hours of Lecture or Seminar per week: 3

Repeatable: May be only taken once for credit, limited to 3 attempts (N3)

Max Allowable Credits: 9

Default Grade Mode: Undergraduate Regular

Recommended Prerequisite(s):

Recommended Corequisite(s):

Required Prerequisite(s) / Corequisite(s) (Updates only):

Registrar's Office Use Only - Required Prerequisite(s)/Corequisite(s):

And/Or	(	Course/Test Code	Min Grade/Score	Academic Level	)	Concurrency?

Registration Restrictions (Updates only):

Registrar's Office Use Only - Registration Restrictions:

Field(s) of Study:

Class(es):

Level(s):

Degree(s):

School(s):

Catalog Description:

Comprehensive introduction to spatial computing using modern computing environments with emphasis on programming and problem solving. Introduces students to modern programming practices using both paradigms of imperative and object-oriented programming. Topics include 1) working with geospatial objects, such as points, lines and polygons; 2) hands-on experience in processing spatial data; 3) solving classic spatial computing problems, such as point-in-polygon tests and line segment intersection tests efficiently; 4) techniques for spatial navigation, such as shortest path algorithms spatial networks; 5) technical challenges such as storing, reading and parsing geospatial data. Tutorials and instruction assume no prior programming experience in Python or other programming languages.

Justification:

The course content have been useful for building computing skills within the domain of spatial analysis. We initially had a version of this course at the 200-level, but have found that students would benefit more from this instruction at the upper-division level. It also dovetails into our existing curriculum that also is offered at the 300-level.

We plan to remove the 200-level version of this course after this proposed version is implemented.

Does this course cover material which crosses into another department? No

Learning Outcomes:

By the end of the course each student will be able to:

- Solve simple problems and tasks using imperative programming
- Solve more complex problems using object-oriented programming
- Have a broad knowledge of data analysis techniques for spatial data.
- Understand and apply basic geometric algorithms.
- Utilize existing packages for advanced spatial analysis and data science
- Articulate and effectively communicate concepts and ideas related to spatial computing to experts, non-experts, and other professionals.
- Have the ability to appropriately apply the knowledge acquired in the course for real-world data.
- Analyze a given dataset in a team

Attach Syllabus

GGS 366 - Syllabus.pdf

Additional Attachments

Staffing:

Current faculty will be reassigned from 200-level

Relationship to Existing Programs:

Will be integrated into course programs after approval, up-leveling current offerings.

Relationship to Existing Courses:

Replacing lower level course

Additional Comments:

Reviewer Comments

Nathan Burtch (nburtch) (02/02/21 4:05 pm): Rollback: Comments from the GGS CC (emailed to AZ and TL)

Key: 17093

Department of Geography and Geoinformation Science
4400 University Drive, MS 6C3, Fairfax, Virginia 22030
Phone: 703-993-1210, Fax: 703-993-9299
Email: ggs@gmu.edu Web: ggs.gmu.edu

GGS 366
Spatial Computing

1. General Information

Instructor:	Dr. Andreas Züfle
Where:	Online/Blackboard
When:	Lectures: Online – Asynchronous Tutorials: Thursday: 10:30am – 11:45am (Zoom)
Course website:	Blackboard
Credits:	3
Prerequisites:	None
Office Hours:	By appointment only due to social distancing. Just email me! (azufle@gmu.edu)

2. Course Description

Spatial Computing has become a ubiquitous part of modern life. Using Virtual Globes, such as Google Maps and Microsoft Bing Maps, and using hand-held GPS devices integrated in smart phones, our society has benefitted immensely from spatial technology. Using this technology, we always know, precisely, where we are, where nearby points of interest such as restaurants are, and how to reach these points of interest. Large organizations use Spatial Computing for site-selection, asset tracking, facility management, navigation, and logistics. Scientists use GPS to track endangered species to better understand behavior, and farmers use GPS for precision agriculture to increase crop yields while reducing costs.

This course provides a comprehensive introduction to spatial computing using modern computing environemnts with emphasis on programming and problem solving. It introduces students to programming in the programming language Python using both paradigms of imperative and object-oriented programming. Topics include 1) working with geospatial objects, such as points, lines and polygons; 2) hands-on experience in processing spatial data; 3) solving classic spatial computing problems, such as point-in-polygon tests and line segment intersection tests efficiently; 4) techniques for spatial navigation, such as shortest path algorithms spatial networks; 5) technical challenges such as storing, reading and parsing geospatial data.

During the first half of this course, it provides tutorials on programming in Python. These tutorials require nor assume prior programming experience in Python or any other programming language.

3. Learning Outcomes

By the end of the course each student will be able to:

- Solve simple problems and tasks using imperative programming in Python
- Solve more complex problems in Python using object-oriented programming
- Have a broad knowledge of data analysis techniques for spatial data.
- Understand and apply basic geometric algorithms.
- Have knowledge about data analysis techniques such as clustering and classification.
- utilize existing Python packages for advanced spatial analysis and data science for your future job and research.
- Articulate and effectively communicate concepts and ideas related to spatial computing to experts, non-experts, and other professionals in a work environment.
- Have the ability to appropriately apply the knowledge acquired in the course for real-world data.
- Analyze a given dataset in a team

4. Format

This is an online course, taught as a combination of asynchronous videos and live (but virtually on Zoom) tutorials.

Lectures: Course materials will be presented asynchronously as online-videos on Blackboard.

Tutorials: To learn Python, we will have many tutorial sessions in which you will apply what you have learned in lectures to solve small programming problems to prepare you for assignments and exams. These sessions will be held synchronously on Zoom. For students who are not able to attend the tutorials in person, these tutorials will be recorded and made available on Blackboard. Students can also submit their questions asynchronously to be answered in the tutorials. For example, students can send in their Python code with questions, and these questions will be answered in the tutorial session. For tutorials, it is highly recommended to attend in person, as that allows you to collaborate and discuss with other students, which is simply more fun than solving problems on your own.

Prior Program Experience: *No prior programming experience is required for GGS 366.*
Some of you may already have prior programming experience, in Python or otherwise, and may already have some understanding of the concepts taught in the first half of the course. For those of you, I will have a list of more advanced programming problems, which are entirely ungraded and just to keep you entertained while we cover the basics in class. Your solutions to these advanced problems can be submitted for me to give you feedback but have no effect on grading.
Students with prior programming background are welcome to attend in-class tutorials to help classmates who are new to programming.

5. Textbooks

None. All required reading materials will be posted on Blackboard as course videos. Additional optional material will be provided as links to open Python resources (such as online tutorials).

6. Technology Requirements Hardware

You will need access to a Windows or Macintosh computer with at least 2 GB of RAM and to a fast, reliable broadband Internet connection (e.g., cable, DSL).
For the amount of computer hard disk space required to submit your assignments online, consider and allow for the space needed to save your course assignments.

Software

We will use Python. Python is free to download. We will use integrated development environments (IDEs) that are free to download.

7. Course Outline (tentative)

The first half of the semester we will focus on learning Python. This will include in-class tutorials to make you familiar with the Python programming language. First, you will learn basic (imperative) Python programming. Then, we will move to object-oriented programming to work with spatial objects such as points and polygons. To show your understanding, and exercise beyond the classroom, you will be required to solve programming assignments.
As the course does not expect you to have any programming experience, some of you may already have programming experience. To keep those of you interested, I will make a large collection of additional (ungraded) programming assignments available on wide scale of difficulty levels. The additional assignments won't be graded – but I will give you feedback to help you improve.
The first half of the semester will be wrapped up by a mid-term exam, which will require programming in Python to use solve spatial computing tasks. Most questions on the midterm exam will require programming.

In the second half of the semester, we will use our newly acquired Python programming skills to solve spatial computing programs, including problems in computational geometry (such as the point-in-polygon test), graph problems (such as the shortest path search problem), and data mining problems (such as clustering). The course will be capstoned by your choice of either a final exam or a course project. Course projects can be done in teams, and the level of complexity of the project will depend on the number of team members. Solo projects are also allowed.
Note that a course project usually takes much more time than preparing for an exam but also be more fun. Students that decide to do a course project will be allowed to switch back to the final exam option at any time. In my experience, about 75% of students choose the final exam

A detailed schedule of the topics covered in this course is given in the following (please note that the topics and their order are subjected to change at the discretion of the instructor, any changes will be announced on Blackboard in time):

Week #	**Topic**	**Assignment**
1	Motivation: Spatial Computing, Big Spatial Data and Data Science Intro to Spatial Data: Points, Lines, Polygons. Examples and applications.	Assignment 1: Introduction & Hello World (no points)
2	Intro to Programming in Python: Imperative Programming Part 1 (variables, branching, functions)	Assignment 2: Imperative Programming
3	Intro to Programming in Python: Imperative Programming Part 2 (recursion, iteration)	Assignment 3: Recursion and Iteration
4	Intro to Programming in Python: Imperative Programming Part 3 (Lists and Data Structures)	Assignment 4: Lists and Data Structures
5	Intro to Programming in Python: Object Oriented Programming Part 1 (Basics, Points, Lines)	Assignment 5: Object Oriented Programming
6	Intro to Programming in Python: Object Oriented Programming Part 2 Geometric Algorithms Part 1	Assignment 6: Point in Polygon Test
7	Assignment Discussion and Midterm Q&A	
8	Midterm Exam	
9	Geometric Algorithms Part 2	Assignment 7: Geometric Algorithms
10	Network Shortest Path Search: Shortest Path Algorithms	Assignment 7: Shortest Path Search
11	Path and Motion Planning Algorithms: Collision-free shortest path finding	Assignment 8: Collision-Free Path Search
12	Introduction to Spatial Data Mining Part 1 Overview and Clustering	Assignment 9: Clustering

13	Introduction to Spatial Data Mining Part 2 Classification, Machine Learning and Artificial Intelligence	Assignment 10: Classification
14	Project Discussions and Final Exam Q&A	
15	Reading Days. No Class.	
16	Final Exam and Final Project Presentations	

8. Grades

The midterm exam and final exam (or project) each count for 50% of the total grade.
Assignment points yield ***bonus*** points of up to 20% of the total score. Bonus points will only be applied if the total grade (from exams/project) is better than an F.

Intermediate assignments	**20% (Bonus)**
Midterm Exam	**50%**
Final Exam (or Project)	**50%**

Grades will be based on the following cut of values, although I reserve the right to alter the values at the end of the course: A+ (100%), A (93%), A- (90%), B+ (87%), B (83%), B- (80%), C+ (77%), C (73%), C- (70%), D (60%).

Final grades at the end of the course will be assigned using **ONLY absolute achievements** not considering relative standing in the class. (there is no "quota" for grades)

The way I handle bonus points always causes confusion, so here are a few examples that should answer most questions:

Alice had 80% score in the midterm, and 70% score in the final exam. Her average is 75% which would be a C. However, she also submitted her assignments and got 90% on the assignments score. Since her grade (without assignment bonus) is better than an F, she gets a bonus of 18% (90% of the 20% Assignment Bonus Points). Her final score is 75%+18%=93% which is (barely) an A.

Mike bombed the midterm and only got 50% of the midterm score. He did much better in the final exam and got 80%, and he got 80% of the assignments. His total score without assignment points would be (50%+80%)/2=65%, which is a D. Since D is better than F, he gets his assignment bonus which brings him up to 81% for a B-.

Sarah had 95% in the midterm and 90% in the final exam. She also had 80% of the assignments. Her total from the exams is 92.5%. Assignments give her an addition 16% for a total of 108.5%. This will be rounded down to 100% (as Blackboard does not allow one to have more than 100%) and result in an A+.

Andy got 60% in the midterm and 50% in the final exam. He had 100% of all assignment points. His average from exams is 55%. That's an F. No bonus points are applied to an F, so his final grade is an F.

Will got 100% in the midterm and 100% in the assignments. The final exam is tomorrow, and he asks:
How many points to I need to pass the course? How many points to I need for an A+?
Answer: Will needs 20% in the final exam to pass (to get an average of 60% (D) from the exams – in this case he would get a B-), and he needs 60% for an A+ (that would give him an 80% average from exams plus 20% from Bonus Points).

Let me know if you still have questions about how grading works, how bonus points are applied, or if you want to make sure that your math is right for your own grade!

9. Exams

The course includes a mandatory written mid-term and final exam. The material covered in the exams will be announced in class.

10. Assignments:

The course will include several written assignments on selected topics from the material covered in class and in the assigned reading.

Assignments should be done **through the Blackboard course website**.

Please note: Assignments should be submitted only through the Assignment submission section of the Blackboard system - DO NOT email assignments directly to the instructor.

11. Late Assignment Submission:

Papers submitted **after the due date will not be accepted**. Exceptions to this policy may be made given serious circumstances at the discretion of the Instructor.

Please note: Deferral of term work is a privilege and not a right; there is no guarantee that a deferral will be granted. Please make sure you notify the instructor as soon as you know a deferral is required.

12. General guidelines for ASSIGNMENT preparation and submission

a. Grades of assignments will be based on:

- **Academic merit** of your answers.
- **Conciseness** and **completeness** of your answers. Please write to the point and explicitly address the question or task. Avoid using unnecessary graphics (figures, tables, graphs etc.) unless they serve a specific purpose. Make sure to use captions and to refer to the graphics you include in your written answer. Graphics without any reference or accompanying explanation will be disregarded.
- **Organization** and **presentation**. Remember that your assignment report is a reflection of your thinking and learning process. Please organize your report in a logical fashion so that your answers could be easily identified. A general format for your presentation should, as a minimum, include the following components: (1) Question number, (2) Your written answer and/or description and discussion of your results, and (3) Visualization of your results, e.g. images, graphs, tables, as necessary.

b. Please remember that your assignment is a **professional document**, and should therefore be formatted and constructed accordingly. All assignments are to be typed. Hand-written assignments will not be accepted.

c. Submission of a hardcopy will be made in class; submission of a softcopy will be made through Blackboard.

d. The electronic submission of your assignment report has to be in **PDF format**.

e. If more than one file is submitted, you may submit a single **ZIP** file containing all the assignment files.

f. Each assignment submission should include a cover page with the following information: assignment title, assignment number, student name, and submission date.

g. Please make sure you have a backup of all the materials you submit.

13. Course website:

The course has a Blackboard website. This website will provide you a single portal through which you may obtain lecture notes, retrieve assignment data and, review links to additional materials, and receive special announcements. You are required to visit the course website **once per day**. Please notify ITU (and, if necessary, the instructor) if you encounter any problems accessing this website.

14. Electronic communication:

All course related email correspondence, including submission of assignments, should be made through the course Blackboard website. You also email the instructor direct at azufle@gmu.edu. In that case, please include "GGS 366" in the subject of your email

15. Safe Return to Campus

All students taking courses with a face-to-face component are required to take Safe Return to Campus Training prior to visiting campus. Training is available in Blackboard (https://mymason.gmu.edu). Students are required to follow the university's public health and safety precautions and procedures outlined on the university Safe Return to Campus webpage (www2.gmu.edu/safe-return-plan). Similarly, all students in face to face and hybrid courses must also complete the Mason COVID Health Check daily, seven days a week. The COVID Health Check system uses a color code system and students will receive either a Green, Yellow, or Red email response. Only students who receive a "green" notification are permitted to attend courses with a face-to-face component. If you suspect that you are sick or have been directed to self-isolate, please quarantine or get testing. Faculty are allowed to ask you to show them that you have received a Green email and are thereby permitted to be in class.

16. Student Expectations:

- **Academic Integrity:** Students must be responsible for their own work, and students and faculty must take on the responsibility of dealing explicitly with violations. The tenet must be a foundation of our university culture. [See http://academicintegrity.gmu.edu/distance].
- **Honor Code:** Students must adhere to the guidelines of the George Mason University Honor Code [See http://oai.gmu.edu/the-mason-honor-code/].
- **MasonLive/Email (GMU Email):** Students are responsible for the content of university communications sent to their George Mason University email account and are required to activate their account and check it regularly. All communication from the university, college, school, and program will be sent to students solely through their Mason email account.
  [See https://masonlivelogin.gmu.edu].
- **Patriot Pass:** Once you sign up for your Patriot Pass, your passwords will be synchronized, and you will use your Patriot Pass username and password to log in to the following systems: Blackboard, University Libraries, MasonLive, myMason, Patriot Web, Virtual Computing Lab, and WEMS. [See https://password.gmu.edu/index.jsp].
- **University Policies:** Students must follow the university policies.
  [See http://universitypolicy.gmu.edu].Responsible Use of Computing - Students must follow the university policy for Responsible Use of Computing.
  [See http://universitypolicy.gmu.edu/policies/responsible-use-of-computing].
- **University Calendar:** Details regarding the current Academic Calendar.
  [See http://registrar.gmu.edu/calendars/index.html].
- **Students with Disabilities:** Students with disabilities who seek accommodations in a course must be registered with the George Mason University Office of Disability Services (ODS) and inform their instructor, in writing, at the beginning of the semester [See http://ods.gmu.edu].
- Students are expected to follow courteous Internet etiquette at all times; see http://www.albion.com/netiquette/corerules.html for more information regarding these expectations.

17. Student Services:

- **University Libraries:** University Libraries provides resources for distance students.
  [See http://library.gmu.edu/distance and http://infoguides.gmu.edu/distance_students].
- **Writing Center:** The George Mason University Writing Center staff provides a variety of resources and services (e.g., tutoring, workshops, writing guides, handbooks) intended to support students as they work to construct and share knowledge through writing. [See http://writingcenter.gmu.edu]. You can now sign up for an Online Writing Lab (OWL) session just like you sign up for a face-to-face session in

the Writing Center, which means YOU set the date and time of the appointment! Learn more about the Online Writing Lab (OWL).

- **Counseling and Psychological Services:** The George Mason University Counseling and Psychological Services (CAPS) staff consists of professional counseling and clinical psychologists, social workers, and counselors who offer a wide range of services (e.g., individual and group counseling, workshops and outreach programs) to enhance students' personal experience and academic performance [See http://caps.gmu.edu].
- **Family Educational Rights and Privacy Act (FERPA):** The Family Educational Rights and Privacy Act of 1974 (FERPA), also known as the "Buckley Amendment," is a federal law that gives protection to student educational records and provides students with certain rights. [See http://registrar.gmu.edu/privacy].

Disclaimer: Any typographical errors in this Course Outline are subject to change and will be announced in class. The date of the final examination is set by the Registrar and takes precedence over the final examination date reported by the instructor. **Note**: Recording is permitted only with the prior written consent of the professor or if recording is part of an approved accommodation plan.