COP4521: Secure Parallel and Distributed Computing with Python
Fall 2021

Course Instructor

Alan Kuhnle
Office: MCH 106A
Email: kuhnle@cs.fsu.edu

Meeting Time and Place

MWF 1:20–2:10 pm, HWC 2401. Lecture delivery is face-to-face. All lecture slides will be posted on the course webpage.

Course Webpage

`https://www.alankuhnle.com/teaching/fa21-cop4521/cop4521.html`

Prerequisites

This course is an advanced course in programming. Students are expected to be competent with the programming material that is taught in COP4530 (Data Structures, Algorithms and Generic Programming).

All students taking COP 4521 are required to have previously taken and passed (with a C- or higher final grade) COP 4530 or an equivalent course.

Course Description

Computer Science is a rapidly evolving field and several current problems in the field involve issues of keeping information secure and simultaneously making software run faster and more efficiently. Software Engineering now requires a thorough understanding of secure, parallel and distributed computing.

Python is a very popular and versatile programming language with applications across a variety of domains. It owes its popularity to its ease of use and a large and dynamic list of libraries. This course will examine and demonstrate the principles of secure, parallel and distributed computing and their application in software engineering using the Python Programming Language.

This is NOT a Python Programming class. This is a class that introduces the concepts of information management, computer security, parallel and distributed computing, using Python as the programming language of choice.

Course Objectives

Topics will include lectures on the Python language and development environment as well as coverage of the basic concepts of computer security, parallel and distributed computing, and information management. Upon completion of the course, students would be able to

- Solve simple programming problems using Python.

- Build small scale real word applications using the third party Python libraries discussed in the course.
- Solve problems using concepts of information management, computer security, parallel and distributed computing.
- Apply Python towards several contemporary programming requirements and techniques; specifically - secure, distributed and parallel computing.

Course Schedule

This is a tentative course schedule, which lists the topics and Python libraries by week.

- Week 1: Syntax for Python Programming: Modules, Functions and Object Oriented Programming
- Week 2: Syntax for Python Programming: Data Structures, Standard Library
- Week 3: Introduction to information Management, Databases and SQL
- Week 4: Web Programming and Interacting with Databases using flask libraries, NOSQL Databases
- Week 5 : Introduction to Computer Security and secure coding practices
- Week 6: Access Control, Hashing, and Cryptographic Random Numbers
- Weeks 7,8: Cryptography: Symmetric and Asymmetric Encryption
- Week 8, second class period: Midterm
- Week 9: Malicious Software and Defences, Computationally Secure Languages and Software
- Week 10:Introduction to Networking and Socket Programming
- Week 11: Numeric and Scientific Programming: Numpy, Scipy and Matplotlib
- Week 12: Introduction to Parallel Computing
- Week 13: Parallel Programming using : process, pool, multiprocessing, and threads libraries
- Week 14: Introduction to Distributed Programming
- Week 15: Distributed Programming Using Hadoop
- Week 16: Final Exam - during Finals Week

Textbooks

Textbooks required for the course:

1. Python in a Nutshell, 3rd edition, Author: Alex Martelli, Anna Ravenscroft, Steve Holden.
2. Information Security: Principles and Practice 2nd Edition by Mark Stamp (Wiley)

These textbooks are available from several online retailers. Students will also receive handouts on topics in Parallel and Distributed Computing, Concurrency and Information Management.

Office Hours

The instructor and the Teaching Assistants will hold office hours (both in-person and virtual) every week. For the schedule, see the course webpage.

Software Required

For this course, we would be using Python 3.8 on a Ubuntu machine. Both are available for free. It is recommended that you use a Virtual Machine (like VirtualBox, also available as free software) or a Virtual Environment, in order to leave the original settings of your computer undisturbed.

Assignmetns, Projects, and Tests

Assignments will be given periodically through the semester. They will be posted on the course website. Students will have a week to 10 days to complete these assignments.

- Assignments are NOT OPTIONAL. Students need to turn in all the homeworks to make an attempt at getting full credit for the homework/assignment component of the grade.
- Students are expected to turn in all assignments ON TIME!
- Students are not permitted to "re-do" assignments after the deadline.
- Assignment deadlines are STRICTLY enforced.
- STUDENTS are responsible for ensuring that their program file was submitted correctly. This means making sure their file was submitted without error, ON TIME, and also submitting the correct .py or .tar file.
- STUDENTS are responsible for ensuring they do not accidentally delete or overwrite their files.
- Interpretation Errors
  - Programs that do not run are very tedious to grade, and they show a lack of testing, which is a large part of programming. There will be an automatic 5 point penalty for each interpretation error in a student's code that has to be fixed in the grading process. (This means that program submissions with interpretation errors will likely earn very little, if any, credit).
  - If there are more than 10 interpretation errors the program receives an automatic zero. Students are responsible for making sure the code RUNS before they submit it.

There will be two tests over the course of the term. The tentative date for the midterm is the second class period of Week 8. The final will during the scheduled final timeslot during finals week.

The tests will be cumulative. The test format will be a mixture of multiple choice, short-answer, code reading and understanding, and code writing.

Group Project

The course project is a semester-long project which will be assigned towards the beginning of the course. Students must work in groups of 3 or more. The course project should involve interaction with a database and apply the principles of information management. Students will be required to submit a proposal in the beginning of the semester, but the topic choice is completely open-ended. All project work will be done using a code repository like github. The instructor will closely monitor the student's individual contributions to the project. The grade for the project will be based on:

1. Overall functionality.
2. Whether all of the requirements of the original proposal were met.
3. The size and quality of the student's individual contribution to the project.
4. Overall quality of code. (code organization, coverage, complexity, test system, build system, documentation).
5. Demonstrating the ability to form a team and work with your team members to produce an integrated, cohesive software artifact.

The group is also expected to present a short demonstration of the project for the course at the end of the semester.

Attendance

Attendance is not required. The lecture slides for each class will be posted on the course webpage.

Practice Exercises

Practice is the only way to get better at programming. Programming is a very incremental discipline, and material covered in one week will be used in all the subsequent weeks. It is recommended that students practice for about 30 minutes every day. Students will be given ungraded practice exercises, posted on the course website. It is recommended that the students try and solve these practice exercises and ask the instructor or teaching assistants for help if they encounter issues.

Grading Policy

The final course grade will be computed as follows:

- Assignments 40
- Group Project 20
- Midterm 20
- Final 20

Requests for regrading should be within a week of grades being posted on Canvas.

Final numerical scores will be converted to letter grades for the course according to the following grading scale.

- 93% – 100%: A
- 90% – 92.99%: A-
- 87% – 89.99%: B+
- 83% – 86.99%: B
- 80% – 82.99%: B-
- 77% – 79.99%: C+
- 73% – 76.99%: C
- 70% – 72.99%: C-
- 67% – 69.99%: D+
- 63% – 66.99%: D
- 60% – 62.99%: D-
- 0% – 59.99%: F

THE CLASS WILL NOT BE GRADED ON A CURVE. THE GRADES WILL NOT BE ROUNDED TO THE NEXT WHOLE NUMBER.

In addition to the scale listed above, in order to earn a C- or better in the course, a student is required to achieve a test average of C- or better. If the test average is below this level, the highest possible course grade is a D+. The test average can be computed with the following formula: TestAvg = ((Midterm * 15) + (FinalExam * 15)) / 30

Canvas only takes the graded assignments into account while calculating your letter grade. Ungraded assignments are accounted as 100%, instead of 0. So, Canvas might show an expected letter grade of A- one day and C- the next. Students are requested to calculate their grade according to the grade distribution, with a 0 for all the grade that haven't yet been posted. If a student needs an Excel formula for their grade, they can email the instructor/ TA's for one.

Course Policies

University Attendance Policy Excused absences include documented illness, deaths in the family and other documented crises, call to active military duty or jury duty, religious holy days, and official University activities. These absences will be accommodated in a way that does not arbitrarily penalize students who have a valid excuse. Consideration will also be given to students whose dependent children experience serious illness.

Late Assignment Policy Students are expected to turn their assignments in on or before the due date. Late assignments will suffer a 10 percentage point penalty for the first 24 hour period. For example, an assignment worth 200 points turned in late will receive a 20 point penalty. Assignments turned in more than a day after the due date will receive a grade of '0', but students can still receive feedback.

Extra Credit Policy Extra credit points will be offered on both the midterm and the final. Students will also get the opportunity to turn in an extra credit homework. Extra credit might be offered for participating in the ACM programming contest (if the contest is organized for Fall 2021) and using Python to solve at least one problem.

Incomplete Grade (Grade of "I") Policy Incomplete ("I") grades should be recorded only in exceptional cases when a student, who has completed a substantial portion of the course and who is otherwise passing, is unable to complete a well-defined portion of a course for reasons beyond the student's control. Students in these circumstances must petition the instructor and should be prepared to present documentation that substantiates their case.

Academic Honor Policy The Florida State University Academic Honor Policy outlines the University's expectations for the integrity of students' academic work, the procedures for resolving alleged violations of those expectations, and the rights and responsibilities of students and faculty members throughout the process. Students are responsible for reading the Academic Honor Policy and for living up to their pledge to "...be honest and truthful and...[to] strive for personal and institutional integrity at Florida State University." (Florida State University Academic Honor Policy, found at `http://fda.fsu.edu/Academics/Academic-Honor-Policy`).

Additional Notes on the Academic Honor Policy In addition to the University's Academic Honor Policy, students are expected to be aware of the following:

- Students are expected to do their own work on any homework or test submitted for a grade. The group project is also graded on a student's individual contribution to the project.
- It is NOT appropriate to work on assignments with other students or to give or receive solutions to or from anyone before an assignment is due and handed in (by all parties).
- It is NOT appropriate to share any amount of code with your classmates.
- Using or submitting existing programs/reports from the internet is a violation of the Academic Honor Policy.
- Submitting programs/reports/assignments done, wholly or in part, by a third party, including hired and contracted is a violation of the Academic Honor Policy.
- DO NOT POST YOUR CODE ONLINE (online compilers, text sites, blogs, help sites, etc...). No matter what the intent was in posting your code, this is automatically in violation of the Academic Honor Policy and the appropriate actions will be taken. DO NOT USE online compilers, chat rooms, or post any amount of your code on the web. The only exception is the code for the group project, which is expected to be posted in a private repository.
- Discussing solutions and techniques on assignments with other students after the assignment has been graded and handed back is okay, and encouraged.

- Students are expected to turn in their work with their name on it, and they are representing that work as their own. If a student's submission matches that of another student, it is considered a violation of the Academic Honor Code.
- If a student has previously taken the course, they are NOT permitted to submit their old work for any assignment in the current semester. They must do their work from scratch. This is included in the FSU honor policy. See the link above.
- If it is found that a student has violated the academic honor policy the student is not permitted to drop or withdraw from the course, and must complete the course with the sanctions accessed via the policy. This is a UNIVERSITY policy.
- Examples found in the course textbook or in the course notes may be used in programs, as long as the source is cited. This is appropriate, as some hand-in assignments may be based on program examples found in the book or contain other code that is provided to you in the assignment specification.
- A first violation of the honor code will result, at minimum (but not limited to), a penalty of a 0 grade on the assignment or test involved, along with a reduced letter grade in the course. This will be done by filing the Step-1 Agreement of the FSU Honor Policy.
- Any second violation of the honor code will result in an automatic F in the course, and possible proceedings before the Honor Court. This will be done with a Step-2 Hearing.

Americans with Disabilities Act Download PDF Version of ADA

Florida State University (FSU) values diversity and inclusion; we are committed to a climate of mutual respect and full participation. Our goal is to create learning environments that are usable, equitable, inclusive, and welcoming. FSU is committed to providing reasonable accommodations for all persons with disabilities in a manner that is consistent with academic standards of the course while empowering the student to meet integral requirements of the course.

To receive academic accommodations, a student:

(1) must register with and provide documentation to the Office of Accessibility Services (OAS); (2) must provide a letter from OAS to the instructor indicating the need for accommodation and what type; and, (3) should communicate with the instructor, as needed, to discuss recommended accommodations. A request for a meeting may be initiated by the student or the instructor.

Please note that instructors are not allowed to provide classroom accommodations to a student until appropriate verification from the Office of Accessibility Services has been provided.

This syllabus and other class materials are available in alternative format upon request.

For more information about services available to FSU students with disabilities, contact the
`Office of Accessibility Services 874 Traditions Way 108 Student Services Building Florida State University Tallahassee, FL 32306-4167 (850) 644-9566 (voice) (850) 644-8504 (TDD) oas@fsu.edu https://dsst.fsu.edu/oas`

Syllabus Change Policy This syllabus is a guide for the course and is subject to change with advanced notice. Changes to this syllabus must be accomplished in writing and posted to the appropriate sites.