

Course Syllabus - COMP120 (fa23)

Programming Abstractions and Methodologies

Course Instructor

Dr. Sat Garcia¹

- **Office:** Guadalupe 104
- **Email**²: sat@san Diego.edu

Course Meeting Times

	Days	Time
Lectures	MWF	10:10 - 11:05am
Labs	M	2:30 - 3:50pm

Office Hours

Schedule

Day	Time	Mode
Monday	4 - 5pm	In-Person
Tuesday	10am - Noon	Remote
Wednesday	4 - 5pm	In-Person
Thursday	<i>None</i>	-
Friday	3 - 4pm	In-Person

I also welcome individual meetings by appointment. Send me a DM or talk to me in class to schedule a time.

Locations:

- **In Person:** Guadalupe 104
- **Remote (Zoom):** [Dr. Sat's Meeting Room](#)

¹ Please feel free to call me Dr. Sat or Dr. Garcia. Please don't call me "Professor Garcia."

² I highly prefer to communicate via CampusWire. You are much more likely to quickly get an answer if you send questions through CampusWire. Read onward for details about CampusWire!

Course Goal

Up to this point in your education, you should be familiar with the basic syntax of Python and apply basic patterns (e.g. the accumulator pattern) to solve computational problems. This course will build upon that knowledge to allow you to solve even more complex computational problems from numerous disciplines, such as planning escape routes during emergencies.

A large part of this class will be in learning to manage the complexity of our programs through the use of abstractions and different programming paradigms. As part of this process, we'll learn about interesting data types (e.g. stacks and trees) that make modeling and solving computational problems much easier. We'll also continue to explore how functions and classes can make our programs easier to understand.

Finally, we will start to explore more formal ways of ensuring both the correctness and efficiency of our programs.

Course Learning Outcomes

At the end of this course you should be able to the following.

1. Use abstraction to build programs to solve more complex problems (abstract data types, higher order functions, use of existing libraries).
2. Design and implement computer programs that use object-oriented, functional, and event-driven paradigms.
3. Evaluate trade-offs between different algorithms for solving a problem (space and time trade-offs, big O notation, searching and sorting algorithms as examples).
4. Design and implement an abstract data type.
5. Use modern tools for software development and maintenance (IDE, debugger, source code management).

Tentative Schedule of Course Topics

Below is a list of topics that I expect to cover during the course. It is tentative and subject to change based on how we progress throughout the semester.

1. Python Syntax and Basic Patterns
2. Exceptions
3. Testing
4. Graphical User Interfaces / Event-based Programming
5. Algorithmic Analysis
6. Stacks and Queues
7. Searching and Sorting Algorithms
8. Linked Lists
9. Recursion
10. Hashing

11. Trees

Lab sessions will touch upon the same topics listed above, but also include hands-on introductions to some software tools such as debuggers.

A detailed schedule that will be updated regularly is available [here](#).

Online Course Components

Course Website: [Canvas](#)

Canvas is a replacement for Blackboard that being adopted by all USD courses soon. Our Canvas site will be used to distribute all of the assignments, track grades, and store important course documents. It will remind you of all of the upcoming deadlines so you should check it regularly so you do not miss anything.

Practice Problems: [SpacedCadet](#)

Learning programming is just like any other skill: it requires consistent practice to master. This course uses a website named SpacedCadet as a source for additional, small-scale practice problems. This website has been developed here at USD and is free to use.

Discussion Site: [CampusWire](#)

CampusWire is a fantastic site that enables everyone in class to quickly ask and answer questions. Please make CampusWire your first stop for asking all class-related questions: it is much faster than waiting for me to reply to your emails and there is no better feeling than having the instructor endorse one of your answers to another student's question.

The more that everyone uses CampusWire, the more useful it becomes so please use it regularly. You may be intimidated to post to online forums, especially if you are afraid that your question is "silly." Have no fear though: you won't be judged as there are no silly questions. If you are still nervous, CampusWire has you covered: you can post anonymously.

CampusWire makes it much easier for me to keep track of "Unresolved questions" and therefore I highly prefer all course communication to go through there instead of email. Please don't be offended if I redirect your email questions to CampusWire: most questions are of interest to your classmates as well. If you don't want anyone but me to see your question, CampusWire allows you to make questions/notes as private so that only I can see them. You may also send me a direct message (DM) through CampusWire if you need to chat about anything.

CampusWire also has topical chatrooms, ranging from the serious (chat about your class assignments) to whimsical (share your favorite memes or comics in the meme chatroom). You can also create your own private chat rooms in case you and a group of classmates want to have a safe space to chat together. Unless I am invited, I will not be able to see anything you post in those private rooms.

Neither I nor your classmates will tolerate abusive online behavior so please keep your discussion civil.

Course Materials

Required Textbooks

This course will use two textbooks. Both are free, online and interactive textbooks offered through Runestone Academy. The assigned reading can be accessed directly through our Canvas website: there is no need to sign up for a Runestone account.

1. **Problem Solving with Algorithms and Data Structures in Python, 3rd Ed.**

Bradley N. Miller, and David L. Ranum.

This textbook uses Python to introduce important data structures and algorithms used widely throughout computer science. It is written in an accessible style (i.e. not too dense), without a lot of distracting “extras” that other textbooks tend to include.

2. **How to Think Like a Computer Scientist: Interactive Edition.** *Jeffrey Elkner,*

Bradley N. Miller, and David L. Ranum.

This textbook will be used to supplement the main textbook throughout the semester. It also happens to be the textbook used in COMP110, so most of the content should be familiar to you already.

Classroom Clicker

We will be using clickers to enhance learning during class meeting times. Clickers allow me to instantly receive feedback from you and your classmates, helping me to tailor each class to address misunderstandings. For this class, you may use either the iClicker Student App **or** a physical clicker device (iClicker+ or iClicker2). The app works either as a mobile app (Android and iOS) or as a webapp through any browser. The app has the advantage of being slightly cheaper (\$15 vs \$25 for the semester) but it also makes it easier to get distracted in class, which will harm your learning.

You will receive participation credit for answering clicker questions so please make sure you have the app or a physical clicker available during all classes.

Grading

The university forces me to give out grades at the end of the semester³. I would like to make the grading process as painless as possible and as such I have tried to develop a grading system that is both fair and transparent.

³ This is probably my least favorite part of being a professor.

Your final grade will be a weighted average of the components listed below. Each of these components will be described in more detail throughout the rest of this document. Please make sure you read them carefully. If anything is unclear, please feel free to ask me.

- **Problem Solving Assignments (7):** 30%
- **Final Exam:** 15%
- **Midterm Exams (2):** 30%
- **Quizzes (10):** 10%
- **Class Participation:** 8%
- **Pre-Class Reading Assignments:** 7%

As much as I hate to admit it I do occasionally make grading mistakes. If you happen upon such an error (e.g. incorrect or missing grade), let me know in writing within a week of posting and I will fix the error⁴. After one week the grade will become *immutable*.

Class Participation

The course will be taught using *peer instruction*, a pedagogical technique that relies heavily on active participation by all students during class “lectures.” Class lectures are designed to illuminate common misconceptions with the material and not just to regurgitate what is in the book. Lab sessions will also involve active learning and attendance will be critical to your success on assignments.

Attending every class (lecture **and** lab) and actively participating puts you in a strong position to succeed in this course. Active participation in class will be measured by the number of clicker questions you answer **not** how many you get correct: you will get credit if you answer at least 75% of clicker questions.

To protect everyone’s safety, USD allows instructors to mandate that masks be worn during class. While it is my hope to avoid it, there *may* be some class sessions where I may require that everyone wear masks. During those class sessions, you will be asked to leave class if you do not have a proper mask or refuse to wear one properly so please make sure you have one with you at all times. If you have a battling a cold, the flu, or COVID, please respect the health and wellbeing your classmates and **do not come to class**.

Occasionally events arise that make it so that you cannot attend class. If such an extenuating circumstance arises, please send me a DM on CampusWire before class to let me know you will not be in attendance. You can miss up to **three** lectures and **two** labs without it affecting your participation grade. If you miss more days than that, we will need to chat to make sure you are still in a position to succeed in the class.

CampusWire provides another way to actively participate in class. Through CampusWire you will be able to ask and answer questions, make comments, and upvote posts and responses. CampusWire tracks your participation through a reputation system. To receive

⁴ Please be nice about it: my ego will probably be bruised from realizing I made a mistake.

full participation credit for this course, you will need to reach level 2 in reputation (i.e. Intermediate), which requires you to answer 5 questions on the class feed, and receive 10 upvotes from classmates. You will receive extra credit if you reach the level 3 (Advanced) reputation.

Group Work

You will be working closely with your classmates during class. You should expect them to fully participate in group activities so please offer them the same courtesy. This will require that both you and they perform all the assigned pre-class activities (e.g. readings). You may find that many times you are unsure of the correct answer. This is 100% expected so please don't let it discourage you from letting others know what you are thinking: you likely have a lot more to contribute than you believe.

Working in a group can be intimidating, especially if you feel you are not at the same level as other students. Please remember that the other students are probably feeling the same way: one of my goals in getting you to work in groups is to realize that other students are having similar struggles and that you can overcome them by working together. On the flip side: if one of your classmates seems particularly shy, do your best to support them and encourage them to participate.

Pre-Class Reading Assignments

As part of the *peer instruction* teaching style, I will expect you to come to class having demonstrated proficiency in some basic skills. This will allow us to *actively work* on the most difficult aspects of the material during our meeting times.

To aid in your preparation, for many classes there will be Runestone reading assignments. The reading assignments will ensure that you have read and interacted with the assigned reading. These assignments may also contain a short reading comprehension quiz in addition to the normal interactive reading exercises.

The reading assignments will be due **before** the start of class. While no credit will be given if you complete the reading after this time, you are still encouraged to go back and complete any missed reading assignments as soon as possible.

Quizzes

We will have a quiz approximately once per week. The quizzes will be announced *at least* one day before they occur, but a tentative date for all quizzes is given on our detailed course schedule. The quizzes will give you a chance to test what you have learned over the past week. They will also help you prepare for the exams.

Quizzes are scheduled to take place at the beginning of lecture but depending on circumstances, they may also be administered asynchronously.

As with class participation, I realize that occasionally you will not be able to attend class. Your lowest quiz grade of the semester will be dropped so if you miss one quiz this

semester, it will not count against you. Missing more than one quiz may lead to you falling behind so you should talk to me if that occurs.

Problem Solving Assignments (PSAs)

Problem Solving Assignments (PSAs) give you a chance to go from a problem statement to an implementation of a program that solves that problem. PSAs will be available on the course website, where they will also be submitted. Submitted assignments should be correct and follow the guidelines given with the assignment. You should do your best to submit the PSAs on time: late submissions may be penalized or rejected.

It is very rare that a real world software project will be done alone: keeping with that spirit, most PSAs in this course will require *pair programming*. Pair programming does **not** mean, “You do this assignment and I’ll do the next one.” Pair programming means actively working together on a program in the same room, using the same computer. In pair programming one person is the *driver* while the other is the *navigator*. The driver operates the keyboard and mouse while the navigator checks what the driver is doing and offers input about what to do next. These two roles should switch periodically. Studies have shown that students engaged in pair programming do better in subsequent programming courses. Here is a somewhat [cheesy video](#) that provides an overview on pair programming, including some tips for making it work effectively for you.

Please remember that it is your responsibility to understand all of the code that you submit.

I will pair you with another student; the list of pairings will be available on the course website. You should contact your partner immediately (with a CampusWire DM) to introduce yourself and try to coordinate your schedules and find times to work together. You may also want to create a private CampusWire chatroom for you and your partner, inviting me if you would like. If you do not hear back from your partner or if they tell you they are planning on dropping the course, please contact the instructor immediately. If I don’t hear from you, I will assume you are staying with your initial partner. You should not plan on “going solo:” even if the initial assignments seem easy to you, later assignments will be very difficult without a partner.

Teamwork is a critical skill on which your future employers will evaluate you. You will greatly benefit if you work hard now on learning how to become a good teammate. Being a good teammate includes always showing up on time for planned meetings and adequately preparing for your meetings. I will periodically ask you to evaluate both your and your partner’s teamwork skills to ensure that your group is functioning well. Working with others can be challenging; if an issue arises, you should first bring up the issue with your partner. If that fails to resolve the problem, please contact the instructor. Don’t put off addressing these issues; often a heartfelt talk with your partner as soon as the issue arises can avert many headaches later.

Please remember that you should never share code with anyone except your partner (and even then, sharing should only happen during pair programming assignments). You should feel free to discuss important course/programming concepts with your classmates as long

as you do not give away solutions for an assignment. If you are unsure whether it is OK to discuss a particular topic, please contact the instructor but err on the side of caution until you hear a definitive answer.

Exams

There will be three major exams in this course: two midterm exams and a final exam. The midterm exams are *tentatively* scheduled for Monday, October 2 and Monday, November 6, both during our regularly scheduled meeting time. The final will be in our normal class meeting room and is *definitively* scheduled for Monday, December 18, 11AM - 1PM.

Each exam may cover all the learning objectives up to (and including) the class session before the exam. You will likely be able to use your textbook and possibly written notes on the exams. Unless explicitly authorized in writing by the instructor, you may not receive any kind of external help on an exam.

A serious illness, a death in the family, and other traumatic events are unfortunately part of life. If you contact me within 24 hours with documentation, I will make an arrangement to make up for your missed exam. This arrangement may include taking the exam later or re-weighting other components of your grade to compensate for the missed exam. Otherwise, exams *must* be taken on time.

Grading Scale

Your final grade will be based upon a weighted average of the individual course grading components (i.e. participation, quizzes, exams, etc.). I do not grade on a strict scale, but you will do no worse than the scale shown below for your final, weighted average.

[93 - 100]	A
[90 - 93)	A-
[87 - 90)	B+
[83 - 87)	B
[80 - 83)	B-
[76 - 80)	C+
[69 - 76)	C
[65 - 69)	C-
[55 - 65)	D
[0 - 55)	F

Note that the upper end of each range is non-inclusive. For example, 90% would be considered an A-, not a B+.

If you are taking the pass/fail option, you must receive at least a C- to pass.

Academic Integrity

The Computer Science Department strongly promotes academic integrity. While collaboration is an important part of this course, it is important that you understand what is acceptable and what is unacceptable forms of collaboration.

Quizzes and Exams

Unauthorized collaboration on quizzes and exams is considered a major violation of academic integrity and will be reported as such.

Programming Assignments

Collaboration is normal/expected in the field of computing, with software engineers often reusing or remixing code written by others. However, being an academic setting, there are some important limitations on how you may collaborate with others. Unauthorized collaboration is a form of plagiarism and a violation of USD's academic integrity policy.

Plagiarism is often an act of desperation. While there will be consequences for plagiarism, I will focus on helping students understand the root cause(s) of their actions and how they can address them. If you do engage in plagiarism, you *will* be caught and will be required to meet with me personally. If you enable others to plagiarize, you will also be required to meet with me. Plagiarized submissions will not receive any points, as that would be unfair to your classmates.

Limits on Collaboration

You are welcome and encouraged to help your classmates with fixing errors that they receive when writing their programs, or with explaining how you might solve a problem. For example, it is fine to help your classmates answer the following types of questions.

- "Why is this line of code reporting an error message?"
- "Why isn't my code doing what I expect it to?"
- "Do I need to use a (insert programming feature here) to solve this problem?"
- "We are thinking about solving this problem using one of these two techniques. Which would be best?"
- "Why is the test case saying the program should print this? Shouldn't it print out this other thing?"

The Computer Science Department offers free tutoring for this course throughout the semester. The tutors are a resource to help answer course-related questions like the ones listed above. You should **NOT** expect (or ask) them to generate code or solutions for you.

Remember that any assignment solution that you submit must be the result of your own work, or (in the case of partner-based assignments) the joint work of you and your assigned partner(s). Otherwise, it is considered plagiarism.

The following actions violate our collaborative principles and are therefore considered plagiarism.

- Copying or remixing all or part of the code of someone with whom you have not been explicitly paired by the instructor.
- Copying or remixing all or part of an assignment solution found on the Internet.
- Submitting work that your partner completed without your help, even if you “checked off” on the work before submission.

Please don’t assume you can cheat the system by using someone else’s code then modifying it so that it *appears* different. Every submission is run through sophisticated plagiarism detection software that catches even the most subtle forms of code sharing.

Use of Generative AI

Generative AI programs such as ChatGPT are exciting new tools that have the potential to improve our lives in many ways. Just like any tool though, they can be misused to the detriment of ourselves and others. It is therefore important to understand their limitations and potential uses.

While these systems can write code and answer programming related questions, they are subject to frequent “hallucinations”, resulting in incorrect/made up answers. A recent study⁵ found that ChatGPT incorrectly answered software engineering-related questions over 50% of the time and its responses overly verbose. In other words: don’t trust the programming-related answers that it gives you.

Also note that these generative AI systems often generate code that will be inscrutable to you because it uses advanced syntax or techniques that are not covered in this class. This will make it nearly impossible for you to debug or even integrate into the rest of your code.

If you do use these systems to help with programming assignments, note the following important limitations:

1. You are **required** to indicate which lines of code were either directly or indirectly generated by AI, using comments inside of your code. By indirectly, I mean using AI to generate code and using that code as a template/guide/model for your own code. Failure to cite the use of generative AI is considered plagiarism.
2. You may not use any code that uses Python syntax or modules that was not discussed in class.
3. *No more than* 15% of your submitted code may be AI-generated *or* based on code that was AI-generated.

Disabilities and Learning Differences

Many students have disabilities or learning differences. It is my goal to make sure those people feel fully supported in this class. If you need special accommodations because of one

⁵ [ChatGPT answers more than half of software engineering questions incorrectly](#)

of these, please reach out to the [Disabilities and Learning Resource Center](#) at the beginning of the semester. They have many resources available to help manage your disability and/or learning difference. Most notably they can provide official documentation of your needs so that I can provide the appropriate resources to help you succeed in this class.

Additional Information

The last day to select the pass/fail option is Thursday, November 2. The last day to withdraw from the course *without* a W is Monday, September 11. The last day to withdraw from the course is Wednesday, November 8.

A grade of incomplete will be assigned only if there is a serious, documented reason that prevents you from completing the requirements of the course. Getting a low grade or falling behind is not a sufficient reason.

The only exceptions to the rules regarding no late assignments are extended absences (one week or more) due to verifiable extraordinary circumstances, and absences due to official USD activity travel. In the case of absences due to a USD activity travel, you must give me as soon as it is available a list of your travel dates.

Syllabus Changes

This syllabus is subject to change during the semester. The instructor will make announcements about any non-trivial changes within a day of the changes.

Document Last Updated: October 24, 2023 @ 05:01 PM