CSC 121: Programming & Problem Solving

Davidson College
Fall 2024

Welcome to CSC 121: Programming and Problem Solving. This course is designed to provide a broad introduction to some of the key ideas from the field of computer science, with a particular focus on computational problem-solving using the Python programming language. Starting from the fundamentals, you will gradually solve problems of increasing complexity, by designing and implementing algorithms that use constructs such as lists, control-flow structures, loops, functions, recursion, and objects.

Prerequisites. There are no prerequisites for the course aside from an understanding of basic algebra. In particular, no prior programming experience is assumed.

Class Meeting Information

Mondays/Wednesdays/Fridays from 9:30-10:20 PM.
Classroom: Watson 132

Instructor

Dr. Katy Williams
Office: Watson 232
`kawilliams@davidson.edu`

I will be supported by three Embedded Tutors (ETs), who are shared across all three sections of this course. These are current Davidson Computer Science (CS) majors who have a keen interest in teaching and in sharing their passion for CS with others. These students are available to answer your questions over Slack, in-person, and may assist during class time.

Tara Boehmler	taboehmler@davidson.edu
Ivy Yuan	ivyuan@davidson.edu
Bill Stanton	bistanton@davidson.edu

Course Format

This is a flipped, standards-based course with in-person class meetings. Here's what that means in practice: the course material has been partitioned into a sequence of "modules" with mini lectures for you to watch and work through outside of class. In preparation for class, you will actively watch class videos and often code along with the video. Class time will be devoted to checking and reviewing topic understanding, as well as working on programming assignments that further support learning the material. During class, you will complete a coding warm-up and work through practice problems with a small group of your classmates and with help from the instructor. *To best make use of our class time, you must watch the lectures prior to class.* This will allow you to use the class meeting times to collaborate with your peers so that you can fully benefit from our community.

In addition, the instructional team — myself and the embedded tutors — will hold office hours throughout the week. These are opportunities for you to meet with us to further deepen your understanding of the course material, get help on assignments, etc. *I anticipate that, on average, you will need to dedicate about 12–14 hours per week to successfully complete this course.*

Student-Professor Partnership[1] (or "Why should I go to Office Hours?")

To reach the goals of the course's learning outcomes, we need to work together. I highly encourage you to ask questions in class when all students can benefit from hearing questions and answers. If further clarification is needed, please make use of my office hours–I am fascinated by the course material so it is fun for me to talk about it with you! Office hours are also a good time to talk over questions you have about computer science, graduate school, career options, etc. Should you wish to discuss study strategies for this course and/or others, please come by my office hours. By you attending office hours, it demonstrates to me that you are curious about the course and take initiative to solve problems. It does NOT mean that you are incompetent or incapable of solving problems. Finally, I realize that talking with professors can be a scary, but making use of office hours to have such conversations helps you practice skills needed to interact as a young professional. In other words, it is another way to prepare for post-Davidson life.

[1] Adapted from Dr. Kristi Multhaup.

Office Hours

Office hours are **Tuesdays and Thursdays from 9:00-11:00 AM and 1:00-3:00 PM** (so four hours total each Tuesday and each Thursday). You are welcome to message me on Slack to set up a meeting time outside of these hours; I will try to meet requests.

My office hours will be held in **Watson 232**. The embedded tutors set their own meeting preferences and their office hour schedule and location will be listed on Moodle. I am also happy to meet outside my scheduled hours by appointment — please contact me via Slack to set one up.

Communication

Asynchronous assistance will be offered through the Slack messaging app. You should have already received an email invitation to sign up for the CSC 121 Slack workspace. I encourage you to download and install the app on your computer, and optionally, on your phone. All course staff will be monitoring the Slack workspace, so you are likely to receive a much faster response when asking a question using the app, rather than email. You are encouraged to answer your peers' questions on there as well. **My response hours are between 9:00 AM and 6:30 PM. Do not expect an instantaneous response from me outside of those hours.** The Slack workspace can also be a useful forum for finding or forming study groups, collaborating with your classmates, and simply connecting with us and each other. Feel free to use the #random channel to share snippets from your life with us and the rest of the class!

Course Goals[2]

The primary goal of this course is to equip you with the tools and skills necessary to devise solutions to well-defined computational problems by stepping through the stages of design, implementation, documentation and testing.

Learning Outcomes

Specifically, by the end of the course, you will be able to:

- create algorithms for solving simple problems and express them in English or pseudocode,

[2]Adapted from the ACM 2013 Curricular Guidelines.

- implement, test and debug algorithms for solving simple problems in the Python language,
- analyze and explain the behavior of simple programs involving the fundamental programming constructs: variables, expressions, assignments, I/O, control constructs, functions, parameter passing and recursion,
- identify and describe uses of primitive data types, as well as write programs that use them appropriately,
- modify and expand short programs that use standard conditional and iterative control structures and functions,
- design, implement, test and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, the definition of functions, and parameter passing,
- write a program that uses file I/O to provide persistence across multiple executions,
- choose appropriate conditional and iteration constructs for a given programming task,
- apply the techniques of decomposition to break a program into smaller pieces,
- describe the concept of recursion and give examples of its use,
- identify the base case and the general case of a recursively-defined problem, and
- apply consistent documentation and program style standards that contribute to the readability and maintainability of software.

Accessibility Statement

The college welcomes requests for accommodations related to disability and will grant those that are determined to be reasonable and maintain the integrity of a program or curriculum. To make such a request or to begin a conversation about a possible request, please contact the Office of Academic Access and Disability Resources by emailing AADR@davidson.edu. It is best to submit accommodation requests within the drop/add period; however, requests can be made at any time in the semester. Please keep in mind that accommodations are not retroactive.

Additional Resources for Success

Academic Access and Disability Resources (AADR) offers free academic coaching services to students. If you need help with time management, test taking, studying efficiently, or other academic strategies, a professional staff academic coach or peer academic coach can meet with you once or on an as needed basis. Contact AADR@davidson.edu for more information. Please also stay tuned for Peer Academic Coaching events throughout the semester.

Statement on Diversity and Inclusivity[3]

The embedded tutors and I intend to serve students from diverse backgrounds and perspectives equally in this course and to meet your learning needs both inside and outside of class. We view the diversity that you bring to this class as a resource, strength and benefit. We will continually strive to present materials and activities that are respectful of diversity along a multitude of axes: gender identity, sexuality, disability, age, socioeconomic status, political affiliation, ethnicity, race, nationality, religion, and culture. We value your suggestions, and encourage you to share with us any ideas you may have to help us improve the effectiveness of the course for you personally, or for other students or student groups.

It is our goal to foster an environment in which each class member is able to hear and respect others. If something is said or done by a member of the instructional team or a classmate that is particularly troubling, or causes discomfort or offense, we urge you to consider one of the following courses of action:

- Discuss the situation privately with one or more of the instructors. We are always open to listening to students' experiences, and want to work with you to find acceptable ways to process and address the issue. All such conversations will be kept confidential.
- Discuss the situation with your classmates — chances are there is at least one other student in the class who had a similar response. You may then choose to approach one or more of the instructors as a group, to voice your concerns, if that is a more comfortable setting.
- Complete the anonymous feedback form (https://forms.gle/yVxHYPT25VJPZv4q8) that is linked to from Moodle, to bring the incident to our attention.

[3]From Lynn Hernandez, Behavioral and Social Sciences, School of Public Health, Brown University.

- Notify us of the issue through another source such as your academic advisor, a trusted faculty member, or a peer. If for any reason you do not feel comfortable discussing the issue directly with one of us, we encourage you to seek out another, more comfortable avenue to address the issue, such as the Student Counseling Center or via the Office of the Dean of Students.

Course Resources

Textbook

There is no required textbook for this course and all necessary course materials will be provided by me. However, if you would like a reference text, then I recommend the free, online interactive book How to Think Like a Computer Scientist published by Runestone (https://runestone.academy/ns/books/published/thinkcspy/index.html).

Administration

Course administration will be handled chiefly via Moodle:

`https://moodle.davidson.edu`

Please log on as soon as possible to verify that you are able to access the CSC 121 Moodle page. Course announcements, clarifications, homework assignments, and solutions, handouts, and videos will all be posted here, so it is important that you have access to this page.

Submitting assignments

Assignments will regularly be submitted to Gradescope:

`https://gradescope.com`

I have already added you to the correct section in Gradescope. You should have received an email welcoming you to the course to your 'davidson.edu' email account. *Please notify me immediately if you do not have access to Gradescope.*

Software

An Integrated Development Environment (IDE) is a software tool that allows you to write your own computer programs and run them. In this course, we will use `Thonny` (`https:`

`//thonny.org/`) to develop code in the Python language. The Thonny IDE is available to download for free onto your computer (supports Mac, Windows, Linux, and Chromebooks). Thonny is installed on all of the school computers. You can save your Python files to your OneDrive and access them anywhere – we will review this in class. You may choose to install Thonny on your own computer; however, I will not help you install it during class time.

Course Structure, Expectations, and Grading Policy

The material in this course builds upon itself and it is important to have a solid foundation in the early topics before moving on to more advanced topics. As a result, this course uses a *standards-based* grading system. This means that your grade will be determined by your proficiency in each of the topics over the course of the semester. You will demonstrate your proficiency in topic areas by completing:

- short, reading-concept, in-class pop quizzes,
- in-class coding practices,
- longer programming homework assignments, and
- reviews (i.e. tests).

These quizzes and assignments will be graded on an E-M-R scale (from highest to lowest): Exemplary (E), Meets Expectations (M), and Revision Needed (R). These standards translate to 100%, 80%, and 0%, respectively.

Do not worry if you don't obtain an M or E on an assessment on your first try! We learn by making mistakes, and my goal is to encourage you to learn from yours, not to penalize you when you fall short. Multiple attempts are given for each assignment. Course material is broken into 16 modules and your total grade is calculated out of 100 points:

- You will be given 16 in-class **concept quizzes** (one for each module). You may take up to two retakes by first notifying Dr. Williams of your intent through Slack, then taking the retake through the Quiz Center in Watson (see Moodle). *Once you receive an E on a module quiz, you may not take any more retakes for that module* (not even for "extra practice). Each concept quiz is worth a total of 1.5 points for a total of 24 points towards your total grade.
- You will also be given 16 **coding practices**. These quizzes will completed in class with your teammates. You must upload your code to Gradescope, where I will provide feedback on your style. The goal of these coding practices is to give you practice thinking

algorithmically and writing with good style to help with the homework assignments. I will grade these primarily for completion: you will receive an E for code with minimal style issues, an M of there are multiple style issues, and an R if the code does not pass the autograder or has severe style problems. Each coding practice will be worth 0.75 points for a total of 12 points towards your final grade.

- You will complete 8 **programming homework assignments** throughout the semester, always due **by 11:55 PM on the due date**. These assignments must initially be submitted on their due dates. However, students are highly encouraged to continue working on the assignments even after submission. Students will be given a second submission chance, without penalty, due *no later than one week after grades for that assignment are posted* (or the final day of classes, whichever is sooner). Students are encouraged to resubmit improved versions of their homework assignments. Improved homework submissions will boost final homework grades and will be taken into consideration when determining overall final grades. Each homework will be worth 4 points for a total of 28 points towards your final grade.

- You will complete 3 **reviews** (i.e. tests) that assess materials from each of the 16 modules and 8 programming homework assignments. Reviews will take place in-class. *The third review will be a self-scheduled exam through the Exam Center.* Reviews will test concepts through the modules currently covered in the course. Later reviews will give students second chances to demonstrate understanding on earlier material if an E was not yet achieved during a prior review. The first two reviews are worth 10 points toward your final grade and the third review (the final) is worth 12 points.

In summary, your course grade will be be determined by:

Reviews (10 + 10 + 12)	32%
Homework assignments	32%
Concept quizzes	24%
Coding quizzes	12%

We, the instructor and ETs, are always here to help you. I expect that every student will need help at some point in this course: this is not a cause for embarrassment, it is completely normal. We are here to facilitate your learning, and we encourage you to reach out to us at any time.

Grades

Your final grade for the course is out of 100 points. 32 points from programming homework assignments + 24 points from concept quizzes + 12 points from coding quizzes + 32 points from reviews.

The third review (the final) will be a self-scheduled exam through the Exam Center. You may take your review during the morning or afternoon session on December 12, 13, 16*, or 17 (*The academic calendar currently lists the 16th as an exam day but also closed. If this ends up being an issue, please communicate with me and we will find a solution.*

Your final grade will be determined based on the total points earned during the semester, using the following scale: $A \geq 93\%$, $A- \geq 90\%$, $B+ \geq 87\%$, $B \geq 83\%$, $B- \geq 80\%$, $C+ \geq 77\%$, $C \geq 73\%$, $C- \geq 70\%$, $D+ \geq 67\%$, $D \geq 60\%$, and $F < 60\%$.

Attendance and Collaboration

Please note the following additional policies regarding attending class:

Attendance

This class is designed as a hands-on lab based class where attendance is essential and you should arrive prepared for each session. I assume that every student will need to miss a day or two due to unforeseen circumstances (illness, accident, etc.). If you miss class due to unforeseen circumstances, notify me as soon as possible to devise a plan to make up the missed work. Likewise, if you know you will need to miss class, email me ahead of time and we can discuss accommodations. Note that coming to class unprepared or distracting other students (see Class Participation and Preparation below) is equivalent to not attending class. By Davidson College policy, missing over 25% of class meetings is grounds for receiving a failing grade in the course.

Please look carefully at the syllabus during the first week of class. If any of the due dates conflict with a major religious holiday for your faith, then please let me know. I will make every effort to make the necessary accommodations. Religious observance warrants a legitimately excused absence.

Class Participation and Preparation

It is expected that you come to class prepared and ready to learn. The classroom environment is critical for everyone's ability to learn and we must work together to create a learning

environment. When you are in class, please be respectful of everyone in the room by silencing your phone and putting it away or facedown on your desk. When using the classroom computers, look up only material that is relevant for the class. Other material is distracting both you and those around you. If you need to use your phone for an emergency, please quietly step out of the room. If you see someone with their phone out or irrelevant material on their computer/phone, ask them to put it away.

Collaboration

Learning does not happen in a vacuum, and working with others is often one of the most effective ways to deepen your own understanding of a concept. While the course staff are always ready to work with you, we also strongly encourage you to seek out your classmates and collaborate with them, as you move through the course material. Note that certain assessments — specifically, quizzes and reviews — must be completed solo, without any input from another person. It is a good idea to study for such a quiz or review by first working with another person or two.

Getting Help

Your success in this class depends on many factors — regular attendance, keeping up with the readings, starting early on the homework assignments, etc. But importantly, you also need to work *smart*. This means recognizing when you need help and seeking it out. **I expect that all students will need help at some point in this course.** If you find yourself needing help, this is not a cause for embarrassment — it is completely expected.

There are a number of avenues for getting assistance. **Your primary source of peer-learning support and assistance for this course should be our embedded tutors.** Please visit them in their office hours and ask them questions through Slack. The exact schedule for help sessions and office hours with the embedded tutors will be announced on Moodle. Your other main source of help should be my office hours or through scheduling an individual meeting with me. You will soon find out that office hours are popular in this class and a great way to brainstorm with other students.

Finally, you may use the Internet and search engines to troubleshoot **error messages or syntax questions only**. I will demonstrate appropriate Internet use in class. **You may not directly copy code from another website, student, or generative AI source.**

The *Math & Science Center (MSC)* will also have a small number of additional peer tutors available on a drop-in or by-appointment basis. The embedded, drop-in, and by-appointment tutors are trained and highly qualified peers that demonstrated deep understanding and succeeded in this course themselves. Located in the *Center for Teaching & Learning (CTL)* on the first floor of the College Library, the MSC's drop-in hours are Sunday through Thursday, 8-11 PM, beginning Sunday, September 1. Prior to visiting for drop-in help, be sure to look at the tutor schedules to determine when an appropriate tutor for your course/topic will be present. Tutor schedules for drop-in assistance, as well as links to schedule an appointment with a tutor, can be found at `https://www.davidson.edu/offices-and-services/center-teaching-and-learning/student-resources/math-science-and-economics-center` (click on "Meet with Math or Science Tutor"). Peer assistance is free to Davidson students at the point of service. For more information, contact Dr. Mark Barsoum, Director of the CTL (mabarsoum@davidson.edu or ext. 2796).

Academic Integrity

All homework assignments, reviews and exams in this course are pledged and the Davidson College Honor Code applies:

`http://www.davidson.edu/about/distinctly-davidson/honor-code`

I expect that any work you submit for this course will be your own. At no point should you be in possession of any part of another person's work (current or past students, solutions from Internet resources, or solutions from past semesters) in written or electronic form. *Specifically, unequal collaboration or copying on homework assignments is prohibited.* By "unequal collaboration", I mean one student is clearly providing the majority of thought and code for a homework assignment. Homework assignments are individual submissions; however, you and your peers may have high-level discussions about the assignment and may whiteboard a solution together, similar to how we work on the coding practices. If you ever feel like you are part of an unequal collaboration, either remove yourself from the partnership or ask Dr. Williams to intervene.

In a similar vein, at no time are you allowed to be in possession of code that you did not write yourself. This includes any code obtained from other sources including peers, the internet, or generative AI systems such as ChatGPT. You may not share or receive any part of code with other students. **This includes any student currently taking CSC 121 or any student who has taken CSC 121 in the past.** If you took CSC 121 in the past

and are taking it again, you are not allowed to be in possession of any of your own past work, class notes, or solutions from a past course offering. Cases of cheating or plagiarism may be brought to the Honor Council for review. *If you ever have questions about the bounds of acceptable collaboration, please contact me for clarification.*

You are not allowed to use search engines* to look up any material related to this class. The websites you are allowed to use in this class will be explicitly listed on assignments or on Moodle. If you would like additional materials, you must first contact the instructor to obtain written permission to access outside of class provided materials. Note that everything you need to solve these problems is available through the course material provided on Moodle.

**The ONLY time you are allowed to use a search engine in this class is to look up how to interpret error messages or to recall syntax.*

Course Calendar

The following table outlines the homework and quiz schedule for the course, to help you plan ahead. **These dates may shift slightly throughout the semester.** Quizzes and coding practices will take place during class time. **Homework assignments will be due on Gradescope by 11:55PM on the date below**. An up-to-date schedule of topics will be maintained on the course Moodle page.

Monday	Wednesday	Friday
Aug 26th **1** Module A	28th **2** Module A	30th **3** Module B
Sep 2nd **4** Module B	4th **5** Module C **HW #1 due**	6th **6** Module C

Monday	Wednesday	Friday
9th **7** Module D	11th **8** Module D, cont'd.	13th **9** Module E **HW #2 due**
16th **10** Module E	18th **11** Module E, cont'd.	20th **12** Module F
23rd **13** Module F	25th **14** Module F **HW #3 due**	27th **15** **Review #1**
30th **16** Module G	Oct 2nd **17** Module G, cont'd.	4th **18** Module H
7th **19** Module H	9th **20** Module I **HW #4 due**	11th **No Class (Fall Break)**
14th **21** Module I, cont'd.	16th **22** Module I	18th **23** Module J
21st **24** Module J	23rd **25** Module J, cont'd.	25th **26** Module K

Monday	Wednesday	Friday
28th **27** Module K **HW #5 due**	30th **28** Module K, recap	Nov 1st **29** **Review #2**
4th **30** Module L	6th **31** Module L	8th **32** Module L
11th **33** Module M	13th **34** Module M, cont'd. **HW #6 due**	15th **35** Module N cont'd.
18th **36** Module N	20th **37** Module N	22nd **38** Module N **HW #7 due**
25th **Thanksgiving Break**	27th **Thanksgiving Break**	29th **Thanksgiving Break**
Dec 2nd **39** Module O	4th **40** Module O	6th **41** Module O

Monday	Wednesday	Friday
9th **42** Module P, textbfFRIDAY class schedule, **HW #8 due**	11th **43** Reading Day, **Review #3** taken through the Exam Center during the Final Assessment Period (Dec. 12, 13, 16, 17	13th **44** All exams must be complete by 5:00 PM on Tues., Dec. 17

Review #3 will take place during finals. This means your final "exam" will be the same format as your first two reviews.

Dr. Williams' Three Rules for Success

1. This is hard and I can do it.
2. If I can't do it, I will ask a teammate.
3. If WE can't do it, we will ask an instructor.

Dr. Williams' Goals for the Semester

1. For my students to not be afraid of error messages.
2. For my students to feel more confident in computer-related or technology-related activities.

How to struggle effectively

- **Show your work.** We need proof that you've thought about the problem in order to help you properly. Sketch your thoughts on a piece of paper or put comments or pseudocode in your code.
- **Something is better than nothing**. Even if you don't think you can solve a problem, submit something (e.g. a sketch, an outline of the solution, possible test cases).
- **Use the standards based grading to your advantage.** If you don't get something right away (or need an extension), your extra chance is built-in. Written quizzes are allowed up to two attempts and homework assignments are allowed 1 week of edits to achieve the E standard.