

**General Course Info**

Time: 2a (MW 9:50am-11:00am, F 9:40am-10:40am)  
Location: Olin 310  
Website: <https://www.cs.carleton.edu/faculty/tamert/courses/cs111-s24>  
Slack: <https://join.slack.com/t/carletoncs111-obr7377/signup>

**Instructor Info**

Name: Dr. Tanya Amert ([tamert@carleton.edu](mailto:tamert@carleton.edu))  
Website: <https://www.cs.carleton.edu/faculty/tamert>  
Office: Olin 301D  
Student Hours: Check the [course website](#) for times and locations

**Prefect and Course Staff**

Prefect: Colin James  
Course Staff: Geoffrey Jing and Chloe Simanek

**Textbook and Resources**

Required Textbook: For this course, we will use *Python Programming: An Introduction to Computer Science, 3<sup>rd</sup> Edition*, by John Zelle (ISBN: 978-1-59028-275-5).

All in-class materials and exercises will be available on the course website: <https://www.cs.carleton.edu/faculty/tamert/courses/cs111-s24>. You will be able to use Moodle to submit programming assignments and view the gradebook.

We will use Slack for questions and announcements. You can ask questions about any of the course material on Slack, but you should not post solutions to the assignments.

**Course Description**

This course will introduce you to computer programming and the design of algorithms. By writing programs to solve problems in areas such as image processing, text processing, and simple games, you will learn about recursive and iterative algorithms, complexity analysis, graphics, data representation, software engineering, and object-oriented design.

**Prerequisites**

There are no prerequisites for this course.

## Target Audience

This course serves as a first course in computer science, and as such, no previous programming experience is necessary or assumed.

Students who have received credit for Computer Science 201 or above are not eligible to enroll in Computer Science 111.

## Goals and Key Learning Objectives

By the end of this course, you should be comfortable:

- creating and using functions and objects in programs,
- using decision and iteration structures,
- working with primitive data types, and
- using object-oriented design to create your own data types.

In addition, students should be familiar with algorithms for searching and sorting, able to distinguish their runtime complexities, and able to trace a recursive program.

## Key Dates

First day of class: Monday, March 25<sup>th</sup>

Last day of class: Wednesday, May 29<sup>th</sup>

No class: Monday, April 29<sup>th</sup> (Midterm Break)

Quiz #1: Friday, April 5<sup>th</sup> (first half of class)

Quiz #2: Friday, April 19<sup>th</sup> (first half of class)

Quiz #3: Friday, May 3<sup>rd</sup> (first half of class)

Quiz #4: Friday, May 17<sup>th</sup> (first half of class)

Final exam: Sunday, June 2<sup>nd</sup>, 3:30 p.m. – 6:00 p.m.

## Technology Requirements

In this course, we will use the CS departmental lab machines in Olin 310 and Visual Studio Code with Python 3. You will be able to complete all of the homework assignments on these machines. This will be the supported platform; if you set up VS Code and Python on your own laptop, you are expected to verify that your code executes correctly on the Olin 310 machines before submitting your assignments.

## **Course Components and Grading**

### Course Components

Lessons: Class will usually be a mixture of lecture and small-group exercises. Students are expected to complete assigned readings after each lesson; in class we will introduce concepts, which will be explored further in readings and assignments.

Classwork: Many class periods will include graded in-class exercises (“classwork”) designed to gauge pacing and general understanding of the material. This work will be done on paper, and usually begun on your own and completed with an optional partner. Classwork can be discussed in student hours to make up missed questions.

Assignments: The programming assignments in this course are a fundamental means of student learning. Assignments will typically be released on a Friday and due the following Thursday. Additionally, there will be a small project of students’ own designs near the end of the term.

Quizzes: During the term, there will be four quizzes, each 30 minutes in duration. The final exam will consist of five parts, each meant to take 30 minutes; the first four will be alternate versions of the earlier four in-class quizzes, and the fifth will cover the material since the fourth quiz. The first four parts of the final exam are optional.

### Grading Criteria

<u>Assessment</u>	<u>Count</u>	<u>Percentage</u>
Classwork	~23	10%
Homework Assignments	8	32%
Project	1	8%
Quizzes	5	50%

Your grade for each of the first four quizzes is the maximum of your original quiz grade (from the in-class quiz) and the corresponding (optional) part of the final exam. The fifth quiz occurs only during the final exam, so your fifth quiz score comes only from that part of the final exam. Each quiz corresponds to 10% of the final grade.

There will be eight programming assignments, as well as a final project of students’ own designs. Assignments will be graded both for correctness and style. This means having well-documented (through comments and/or good variable/function names) and easily readable code. Your bottom 2 classwork scores will be dropped (3 if you visit the instructor’s student hours at least once). Unlike classwork and quizzes, most assignment scores are final. There will be the option to resubmit one assignment at the end of the term. To do so, students will need to complete a small writing assignment; details will be announced around midterm break.

### Late Policy

Homework assignments should be submitted to Moodle by the time listed on the assignment webpage. If you do not use a token (explained below), or you do but submit after the updated deadline, late submissions will be penalized 20% within the first 2 hours, 40% within the first 24 hours,<sup>1</sup> or 80% within the first 48 hours. Any work submitted more than 48 hours after its deadline will receive no credit.

---

<sup>1</sup> Mathematically, the late penalty will be a cap on the possible score, e.g., an assignment submitted four hours late and graded to have 85% of the points earned will receive  $\max(85, 100 - 40) = \max(85, 60) = 60$  points.

Each student starts the term with four tokens. These can be used on any homework assignment for any reason, no explanation necessary. Rules for token use:

- One token provides an automatic 24-hour deadline extension. If working with a partner, both partners must spend a token for an extended deadline.
- At most one token can be used on an assignment. To use a token, you should email me by the original deadline (no explanation, just to say that you're using one).
- Some assignments will have optional extensions that count towards additional late days. At most two extra late days can be earned throughout the term.

There is no token use needed to make up missed classwork. (This also means that if you aren't feeling well, please stay home and get some rest!) Classwork grades from the first five weeks of the term will be finalized at the end of Week 6 (Friday May 3<sup>rd</sup> at 4pm). Classwork grades from the last five weeks will be finalized at 4pm on the last day of classes (Wednesday May 29<sup>th</sup>).

### **Course Schedule (tentative, and subject to change)**

Week 1	Algorithm design, basic data types in Python, representing numbers
Week 2	Strings, lists, iteration
Week 2	Quiz 1
Weeks 2-4	Graphics and objects
Weeks 3-4	Control flow
Week 4	Quiz 2
Week 5	Image processing, file I/O
Weeks 5-6	Software design
Week 6	Quiz 3
Weeks 6-8	Object-oriented software design
Week 8	Algorithm design and hard problems
Week 8	Quiz 4
Week 8	Dictionaries
Week 9	Searching, recursion
Week 9-10	Sorting, algorithmic complexity
Final Exam	Quiz 5 (part of final exam)

Note that some topics may be skipped or replaced depending on time constraints. Homework assignments will typically be due every week, and quizzes will be every other Friday. Project details will be announced later in the term.

We will not meet for class on Monday April 29<sup>th</sup> due to Midterm Break.

## Course Policies

### Attendance Policy

Students are expected to attend all class periods. If you must miss class, you are responsible for checking the schedule to determine what you missed and studying the material on your own. If you know of an absence in advance, check with me for how you may best prepare.

Quizzes cannot be taken after the day on which they occur. If you know you must miss class, you may be able to take it in advance; talk to me before your absence. If you're feeling unwell the day of the quiz, we can arrange a separate room for you; if you're too sick to take a quiz, you will need to rely on taking that portion of the final exam.

### Student Hours ("Office Hours")

Student hours, also known as office hours, are a time that you can (and are encouraged to!) stop by to chat with me about anything. This includes questions about material covered in class or the textbook, questions regarding assignments, exam preparation, opinions on WoW expansions, the major in general, board games, or any other topic.

You are welcome to stop by my office during any scheduled office hours; if you have a schedule conflict, send an email and we'll find a different time. You can also visit the departmental lab hours (of our course staff or those for other courses) or prefect sessions for additional help with the course material.

### Classroom Etiquette

In class, you are expected to maintain proper etiquette. This includes arriving on time, not having conversations during lecture, not using your phone in class, etc.

You are, generally, expected to refrain from computer use during regular class periods. However, there will be in-class time devoted to programming practice. You should not be checking email, etc. during these times.

You are expected to take your own notes during class; you should not be taking pictures of the board.

### Collaboration Policy

For homework assignments, you are allowed to work with up to one other student. You may look at each other's code and help with debugging, but *you are expected to type up all of your code yourself*. You may discuss general approaches with other students (ex: "I used a for loop to iterate through the list."), but you should never look at the code of anyone but your partner.

If you worked with someone else or discussed any part of the assignment, you must each list each other's names in comments at the top of the relevant code files or `readme.txt`. See the assignment instructions for more details.

Quizzes are closed-book, closed-note, and will be taken on paper. You cannot collaborate with anyone or any resources to complete your quizzes.

Classwork will also be completed on paper; you can refer to your notes but you should refrain from typing up any of the code and focus on predicting the answer.

You are expected to refrain from using any online sources to generate the solutions to your assignments. The only acceptable online sources are:

- our class website/Moodle/Slack
- official Python documentation (<https://docs.python.org/3/>)

Unacceptable sources for help when writing code or text for this course include using search engines (e.g., Google, Bing), web forums, or generative AI (e.g., ChatGPT). You should refrain from using coding tools such as GitHub Copilot to help with your code; you won't have such tools on quizzes, and the kinds of coding exercises we're covering in this course are foundational enough that you will be expected to know them without looking anything up in any professional computing setting.

You can learn about error messages by searching online. However, before you do so, ask yourself what the error message means. For example, "TypeError: 'int' object is not subscriptable" means you've tried to subscript (e.g., use `[ ]`) with an integer. As always, consider whether a classmate may have encountered this error before; asking on Slack may help others, not just yourself!

## Academic Integrity

It is a violation of the academic integrity to use any tool (e.g., ChatGPT or GitHub Copilot) to generate code or text for your assignments, or to submit assignment code or text written by others. Additionally, you should never be in possession of anyone else's assignment code before the due date, and never look at anyone else's assignment code but your partner's.

If any student is suspected to have violated the academic integrity policy, a report will immediately be made to the Academic Standing Committee, as described at <https://apps.carleton.edu/campus/doc/integrity>. Ask me if you are unsure about what constitutes acceptable collaboration.

## Illness/COVID-19 Policy

Some of the policies stated in this syllabus may need to be modified at times due to COVID-19 or other illnesses. For example, if I have COVID, we may need to have some class periods over Zoom. If you test positive, you should talk to a friend to get notes, and let me know so that we can work around your absences to enable you to still

achieve mastery over the course content. We will work together to make this term a success, and you will be informed about any necessary changes as soon as possible.

## Inclusion

I strive to create an inclusive and respectful classroom that values diversity. Our individual differences enrich and enhance our understanding of one another and of the world around us. This class welcomes the perspectives of all ethnicities, cultures, gender identities, religions, ages, sexual orientations, disabilities, socioeconomic backgrounds, regions, and nationalities.

## Prefect Program

This course has a prefect. The [Prefect Program](#) offers optional collaborative learning sessions for participating classes. Prefect sessions review course concepts and often focus on critical thinking and problem-solving exercises centered on the course material. All the sessions are free and open to all students enrolled in the class. Our course prefect will use email, Moodle, or Slack to inform everyone in the class about upcoming sessions and availability for 1:1 tutoring.

## College Resources

### Accommodations for Students with Disabilities

Carleton College is committed to providing equitable access to learning opportunities for all students. The Office of Accessibility Resources (Henry House, 107 Union Street) is the campus office that collaborates with students who have disabilities to provide and/or arrange reasonable accommodations. If you have, or think you may have, a disability (e.g., mental health, attentional, learning, autism spectrum disorders, chronic health, traumatic brain injury and concussions, vision, hearing, mobility, or speech impairments), please contact [OAR@carleton.edu](mailto:OAR@carleton.edu) to arrange a confidential discussion regarding equitable access and reasonable accommodations.

### Student Well-Being

Your health and well-being should always be your first priority. At Carleton, we have a wide array of resources to support students. It is important to recognize stressors you may be facing, which can be personal, emotional, physical, financial, mental, or academic. Sleep, exercise, and connecting with others can be strategies to help you flourish at Carleton. For more information, check out [Student Health and Counseling](#) (SHAC), the [Office of Health Promotion](#), or the [Office of the Chaplain](#).

### Title IX

Carleton is committed to fostering an environment free of sexual misconduct. Please be aware all Carleton faculty and staff members, with the exception of Chaplains and SHAC staff, are “responsible employees.” Responsible employees are required to

share any information they have regarding incidents of sexual misconduct with the Title IX Coordinator. Carleton's goal is to ensure campus community members are aware of all the options available and have access to the resources they need. If you have questions, please contact Laura Riehle-Merrill, Carleton's Title IX Coordinator, or visit the [Sexual Misconduct Prevention and Response website](#).

## Math Tutoring

The [Math Skills Center](#) supports all Carleton students in any mathematics course they are taking in which they are experiencing difficulty, either with the mathematical concepts or with the mathematical tools needed to succeed in the course. Their mission is to "level the playing field" by giving students who enter Carleton without strong mathematics backgrounds the tools they need to succeed here at Carleton.

## Quantitative Resource Center

The [Quantitative Resource Center](#) (QRC) offers free peer support for students working with numbers in their non-Math/Stats classes. Chat, drop in, or make an appointment with a trained peer tutor for help with graphs, charts, and writing with numbers; Excel, R, and statistical analysis; and math up through Pre-Calculus.

## Writing Center

The Writing Center provides a space staffed with peer writing consultants who can work with you during any stage of the writing process (brainstorming to final proofreading). Hours and more information can be found on the [writing center website](#). You can reserve specific times for conferences by using their [online appointment system](#).

## Support for Students who use English in Addition to Other Languages

If English is not your primary or home language and you believe you might benefit from working regularly with a writing consultant this term, email Melanie Cashin, [Multilingual Writing Coordinator](#), at [mcashin@carleton.edu](mailto:mcashin@carleton.edu). She can arrange once- or twice-a-week meetings between you and a specific writing consultant throughout the term.

## Disclaimer

The instructor reserves the right to make changes to the syllabus, including homework and project due dates, quiz dates, and class cancellations and modality changes, or example due to weather or illness. These changes will be announced as early as possible.