

CS 115 (section C): Introduction to Computer Science, Fall 2018



Prof. Dave Naumann

Email: dnaumann@stevens.edu

Voicemail: 201-216-5608

Office Hours: North Building 309, Tues 10-11, Weds 9-10. and by appointment

Course Overview:

This is an introduction to computer science, with an emphasis on programming (using the Python language). The topics include: design; algorithmic thinking; recursion; object-oriented programming; ethics in computer science; and some basics about computer systems: machine language, interpreters, compilers, and data representation.

Lectures:

Section C:

MWF 10:00 A.M. - 10:50 A.M. Babbio 104

Required Labs:

You must be registered for one of the labs LG, LH, or LI, which run on Thursdays.

Office Hours:

Office hours are held in the North Building – room TBA. The current hours will be posted in Canvas.

Prerequisite:

Substantial exposure in high school to C, C++, Java, or some other major imperative programming language — or else CS 105 or 110.

Students with no prior programming experience are strongly recommended to take CS 105 or 110 first. (For CS majors, it should be CS 110). This course will cover programming in Python, from the beginning, without assuming any specific background — but it covers the basics at an accelerated pace.

Students with a great deal of programming experience and expertise may find this an easy course but still must do the work; and you will probably find we cover several topics that are new for you.

Required textbook:

CS For All (<http://www.cs.hmc.edu/csforall/>), by Christine Alvarado, Zachary Dodds, Geoff Kuenning, and Ran Libeskind-Hadas.

Software:

During the first lab session you will download the Python programming environment.

Warning! Be sure to use the latest version of Python 3.7. Both 2.7 and 3.7 are current branches of the language, but they have different feature sets. Both are actively used in industry, but in this course we'll use 3.7.

Other resources:

We will use various material from the “CS5 Black” course at Harvey Mudd College (www.hmc.edu) (HMC) as well as our own additions. This and the textbook should be all you need. If you want more, the Python Tutorial (<http://docs.python.org/tutorial/>) is a good introduction to the language features, but not an introduction to programming. The Python Standard Library (<http://docs.python.org/library/>) page has comprehensive documentation.

Credits:

Thanks to Christine Alvarado, Zachary Dodds, Geoff Kuenning, and Ran Libeskind-Hadas — faculty in CS at Harvey Mudd (and U.C. San Diego) — for providing support and adapting their course material for our use. Various improvements have been made by Brian Borowski and Dave Naumann.

Coursework and grading:

Computer science centers on programming, which is learned by doing. The main focus of your work will be programming assignments:

- In-class exercises, some of which will be graded (also known as pop quizzes).
- Labs, to be completed and graded during lab. Sometimes you will be required to work in pairs in the lab.
- Homeworks, to be completed on your own time, usually due Wednesday just before midnight. The assignment, and your submission, is via Canvas.

The course score is a weighted average of the following categories.

- homework & in-class assignments 20%
 - Attendance counts as 1 homework assignment
- labs 20%
- first test 10%
- second test 15%
- third test 15%
- final exam 20%

The lowest lab score will be dropped. Homework may include a few extra-credit problems, which can compensate for occasionally missing a pop quiz. Letter grades, with plus and minus, are assigned using the standard scale in Canvas.

ADDITIONAL REQUIREMENT: to pass the course you must average at least 60% on tests, 60% on labs, and 60% homework (including pop quizzes); otherwise your grade will be F.

FINAL OPT-OUT: If your average on the first three tests is at least 85 (no rounding up whatsoever, even if you have 84.9999), you may skip the final exam and the test average

will be used in place of your final exam score when calculating the course grade¹. Otherwise, you MUST take the final exam. In that case, the grade recorded for the final exam will be the **max**(*final exam score, average score on first three tests*). So, it is always in your best interest to study and do as well as possible at all times.

CS 115 Policies

- **You, your instructor, and the teaching assistants are bound by the Stevens Honor System.** Students are responsible for reading and understanding the course policies in these web pages and for announcements made in class and in the course email list.
- You will be permitted to use the textbooks and course notes for programming assignments (homework and labs). During exams, you are not permitted to use notes, books, computing or communication devices unless a different policy is specifically announced by the instructor.
- During lecture and lab sessions please refrain from talking on the phone, excessive texting, or otherwise being impolite. **Professor Naumann reserves the right to confiscate (temporarily) phones and other computing devices, or expel you from the classroom, if you insist on behaving inappropriately in class.**
- After grades are posted, you will have 3 days to inform your grader of a problem. You should also CC your instructor on the email. Do not try to request a grade change after three days, as you should learn from your mistakes in a timely fashion.

No make-up exams, labs, or quizzes

- You must go to your assigned lab session, unless given permission in advance by a TA.
- There are no make-ups for pop quizzes or exams. The only possible exceptions are in the case of death in the student's immediate family (advance notice is required) or near-death experience of the student. Make-ups may be allowed for lab in case of documented illness.

Individual work

Except when groups are explicitly allowed, work must be done individually. You are encouraged to discuss the problems with your classmates but you must not share the details of the solutions. Not by email, not by text message, not by word of mouth, etc. If you are unsure whether you have shared too much, discuss the situation with the TA or instructor; it is your obligation to avoid even the appearance of cheating. We will use Moss (<https://theory.stanford.edu/~aiken/moss/>) on some assignments to verify your code is not too similar to that of other students in the class. If the system indicates a high likelihood of cheating, we will treat it as a violation of the Stevens Honor System.

Late homework

From time to time, all of us have trouble meeting deadlines, and as a near-beginning college student, you will be confronted with many difficult deadlines. But homework doesn't get

¹ The fine print: For this purpose, the first three tests (T1, T2, T3) will be weighted (25%, 37.5%, 37.5%) when calculating the average test score.

easier to do if it's late, and falling behind can snowball. Hence, the following strict policy for homework will be put in place: 2% penalty for each hour past the deadline.

Communication

- As in all of my courses, you are more than welcome to ask me questions as often as you want, and I will always be happy to help.
- The amount of help provided will be directly proportional to the amount of time left before the deadline. Please don't wait until the day before an assignment is due to see me; it'll be too late for me to provide help and too late for you to truly learn the material.
- Please do not ask me at the end of the semester to find creative ways to increase your grade. If you suspect that you are not doing well, come see me to rectify the situation as quickly as possible so that you will have a good grade at the end of the semester.

CS 115 Goals and Assessment

At a high level, the instructor's goals for this course are to introduce you to fundamental concepts of computer science and to help you develop your ability to design, implement, and test programs. Several skills are needed to successfully write programs, including analytical thinking, systematic experimentation, persistence and patience, organization and time management, interpersonal communication, and effective use of reference material (reading technical documentation, searching the web). We focus on algorithmic thinking and problem solving: Analyzing requirements, algorithm design, functions and procedural abstraction, pre- and post-conditions, data abstraction, and invariants. We will emphasize techniques for design, such as data driven programming and object orientation. We will touch briefly on topics that can be studied in advanced courses, including, ranging from tools for testing and secure coding practices to theories encompassing cryptography and the limits of what is computable.

Official Course Outcomes:

encoding	Explain binary encodings used by Python for integers, real numbers, characters, and images.
execution	Demonstrate the dynamic behavior of Python programs that include array access, conditional execution, looping, object reference, and method invocation (including recursive invocation), by showing successive states of a computation.
exceptions	Interpret the information provided by the stack trace of a thrown Python exception.
design	Given a problem description, I am able to sketch a design as pseudocode or flowchart.
coding	Given a design, I am able to implement the design as a Python program.
class	Write a non-trivial instantiable Python class.
state	Explain the use of memory to implement static variables, instance variables, and local variables. Draw the state of the activation stack at any

point in a computation.

inheritance Given a Python class, write a non-trivial extended class.

testing Write a unit test.

Approximate Weekly Schedule (subject to revision):

Week	Topics Covered
1	Elementary concepts of computer programming
2	Simple Python data types, list concept
3	Definition of Python functions, if/then/else concept
4	Recursion on lists
5	Filtering, map/reduce
6	Functions as values
7	Hardware representation of basic data types
8	Assembly language programming using HMMM simulator
9	Iteration
10	Representation of data: atomic vs. composite, mutable vs. immutable
11	Sorting
12	Object oriented programming: class concept
13	Object oriented programming: inheritance
14	Fundamentals of data structures and algorithms

ACADEMIC INTEGRITY

Undergraduate Honor System

Enrollment into the undergraduate class of Stevens Institute of Technology signifies a student's commitment to the Honor System. Accordingly, the provisions of the Stevens Honor System apply to all undergraduate students in coursework and Honor Board proceedings. It is the responsibility of each student to become acquainted with and to uphold the ideals set forth in the Honor System Constitution. More information about the Honor System including the constitution, bylaws, investigative procedures, and the penalty matrix can be found online at <https://web.stevens.edu/honor/>.

The following pledge shall be written in full and signed by every student on all submitted work (including, but not limited to, homework, projects, lab reports, code, quizzes and exams) that is assigned by the course instructor. No work shall be graded unless the pledge is written in full and signed.

"I pledge my honor that I have abided by the Stevens Honor System."

Reporting Honor System Violations

Students who believe a violation of the Honor System has been committed should report it within ten business days of the suspected violation. Students have the option to remain anonymous and can report violations online at <https://www.stevens.edu/honor>.

LEARNING ACCOMODATIONS

Stevens Institute of Technology is dedicated to providing appropriate accommodations to students with documented disabilities. The Office of Disability Services (ODS) works with undergraduate and graduate students with learning disabilities, attention deficit-hyperactivity disorders, physical disabilities, sensory impairments, psychiatric disorders, and other such disabilities in order to help students achieve their academic and personal potential. They facilitate equal access to the educational programs and opportunities offered at Stevens and coordinate reasonable accommodations for eligible students. These services are designed to encourage independence and self-advocacy with support from the ODS staff. The ODS staff will facilitate the provision of accommodations on a case-by-case basis.

Disability Services Confidentiality Policy

Student Disability Files are kept separate from academic files and are stored in a secure location within the Office of Disability Services. The Family Educational Rights Privacy Act (FERPA, 20 U.S.C. 1232g; 34CFR, Part 99) regulates disclosure of disability documentation and records maintained by Stevens Disability Services. According to this act, prior written consent by the student is required before our Disability Services office may release disability documentation or records to anyone. An exception is made in unusual circumstances, such as the case of health and safety emergencies.

For more information about Disability Services and the process to receive accommodations, visit <https://www.stevens.edu/office-disability-services>. If you have any questions please contact: Phillip Gehman, the Director of Disability Services Coordinator at Stevens Institute of Technology at pgehman@stevens.edu or by phone 201-216-3748.

INCLUSIVITY

Name and Pronoun Usage

As this course includes group work and in-class discussion, it is vitally important for us to create an educational environment of inclusion and mutual respect. This includes the ability for all students to have their chosen gender pronoun(s) and chosen name affirmed. If the class roster does not align with your name and/or pronouns, please inform the instructor of the necessary changes.

Inclusion Statement

Stevens Institute of Technology believes that diversity and inclusiveness are essential to excellence in academic discourse and innovation. In this class, the perspective of people of all races, ethnicities, gender expressions and gender identities, religions, sexual orientations, disabilities, socioeconomic backgrounds, and nationalities will be respected and viewed as a resource and benefit throughout the semester. Suggestions to further

diversify class materials and assignments are encouraged. If any course meetings conflict with your religious events, please do not hesitate to reach out to your instructor to make alternative arrangements.

You are expected to treat your instructor and all other participants in the course with courtesy and respect. Disrespectful conduct and harassing statements will not be tolerated and may result in disciplinary actions.