Computer Science 134: Introduction to Computer Science — Spring 2020 — Remote Syllabus

Instructors: Shikha Singh (`shikha@cs.williams.edu`*) and Iris Howley (*`iris@cs.williams.edu`*)*

Virtual Instructor Hours:	See course calendar
Assistants:	Harun Curak, Diego Esparza, Nathan Thimothe, Amelia Chen, Hugo Hua, Mira Sneirson, Jules Walzer-Goldfeld, Minh Phan, Maria Chapman, Caleb Dittmar, Sarah Lyell, Brian Kamau, Yash Mangal, Rachel Nguyen
TA Hours:	See course calendar
Course Text:	Allen Downey's *Think Python, 2ed*, at `greenteapress.com/thinkpython2/thinkpython2.pdf`.
Web resources:	`http://www.cs.williams.edu/~cs134/`
Technical Support:	Lida Doret (`lida@cs.williams.edu`) & Mary Bailey (`mary@cs.williams.edu`).
Lecture:	Posted Sunday, Tuesday, and Thursdays on the Glow course site.

We are surrounded by information. This course introduces fundamental computational concepts for representing and manipulating data. Using the programming language Python, this course explores effective ways to organize and transform information in order to solve problems. Students will learn to design algorithms to search, sort, and manipulate data in application areas like text and image processing, scientific computing, and databases. Programming topics covered include procedural, object-oriented, and functional programming, control structures, structural self-reference, arrays, lists, streams, dictionaries, and data abstraction. This course is appropriate for all students who want to create software and learn computational techniques for manipulating and analyzing data.

Organization. During lecture hours we will typically learn new concepts and problem solving strategies to solve simple problems. While the learning process is initially supported by an online text, we expect a dynamic approach to the class that will allow us to steer lectures in directions of mutual interest. There are be weekly programming lab assignment and homework assignments to supplement the lecture learning.

Work. You are responsible for reading supporting material (*Think Python* (TP)), watching pre-recorded lectures, and pursuing interaction in online Student Help hours as the semester progresses. In addition, some topics may require you to investigate online resources (documentation, tutorials, and the like).

Grading. Your final grade will be determined according to the following:

- Weekly programming (lab) assignment: 50%
- Weekly homework: 20%
- Four quizzes administered through Glow: 30%

The tentative quiz dates are April 17, April 24, May 15, and May 22. We reserve the right to adjust grades by as much as 5% to reflect course participation.

Late Policy. You are expected to turn in all homework assignments by the due date to receive credit. For labs, each student is allowed a total of three late days during the semester, with at most two late days towards any particular lab. A late day gives you a no-questions-asked 24-hour extension. Note that late days are not fractional: there is no such thing as half a late day. You must request a late day in advance on the form located here: `http://bit.ly/s20late`.

Intellectual Property. No part of this course may be reproduced and distributed in any manner without prior permission from the instructors. In particular, no videos recorded as part of this class may be shared with anyone external to the CS134 course.

Community. We embrace diversity. We welcome all students and expect everyone to contribute and support a respectful and welcoming environment. If you have concerns, please share them with us or the college administration.

Students Who Need Accommodations. If formal accommodations need to be made to meet your specific learning or physical abilities, please contact one of us as soon as possible to discuss appropriate accommodations. Please also contact the Director of Accessible Education, Dr. G. L. Wallace (4135974672) or the Dean's office (4135974171). We will work together to ensure this class is as accessible and inclusive as possible.

Mental Health. Students experiencing mental or physical health challenges that are significantly affecting their academic work are encouraged to contact one of us or to speak with a dean. The deans can be reached at 4135974171.

Honor Code. The Honor Code as it applies to non-programming assignments is outlined in the Student Handbook.

For programming assignments in computer science courses, the honor code is interpreted in very specific ways. When a program is assigned, it will be described as a "test" or "laboratory" program. The Honor Code applies to each as follows (unless otherwise specified):

Test Programs. Any assignment designated as a test program is to be treated exactly as a take-home, open-book test. You are allowed to read your textbook, class notes, and any other source approved by your instructor. You may not consult anyone other than your instructor. The instructor encourages the asking of questions, but reserves the right not to answer, just as you would expect during an exam.
Guideline: Any work that is not your own is considered a violation of the Honor Code.

Laboratory Programs. Laboratory programs are expected to be the work of the individual student, designed and coded by him or her alone. Help locating errors and interpreting error messages are allowed, but a student may only receive help in correcting errors of syntax; help in correcting errors of logic is strictly forbidden. In general, if you are taking photos of someone else's screen, looking at someone else's screen, or telling someone else what to type, it is likely your work is no longer the work of an individual student.
Guideline: Assistance in the design or coding of program logic will be considered a violation of the Honor Code.

If you do not understand how the Honor Code applies to a particular assignment, consult your instructor. Students should be aware of the Computer Ethics outlined in the Student Handbook. Violations (including uninvited access to private information and malicious tampering with or theft of computer equipment or software) are subject to disciplinary action.
Guideline: To protect your work dispose of printouts and copies of your work carefully, and avoid leaving your programs on hard disks in labs and other public storage areas.

The Department of Computer Science takes the Honor Code seriously.
Violations are easy to identify and will be dealt with promptly.

The College and Department also have computer usage policies that apply to courses that make use of computers. You can read more about these policies at

`csci.williams.edu/the-cs-honor-code-and-computer-usage-policy`

Tentative Schedule of Topics

Week of	Monday	Lab	Wednesday	Friday
Feb. 3	—		—	1. Hello, world! (TP1)
Feb. 10	2. Expressions (TP2)	I. Python and Gitlab	3. Functions (TP3)	*Winter Carnival*
Feb. 17	4. Conditions (TP5-6)	II. Procedure	5. Iteration (TP7)	6. Lists (TP10)
Feb. 24	7. Strings (TP8-9)	III. Toolbox Building	8. Mutability, Tuples (TP12)	9. Files (TP14)
Mar. 2	10. Sets, Dicts, (TP11)	IV. Faculty Trivia	11. Plotting Data	12. Generators
Mar. 9	13. Iterators	V. Presenting Data	14. Classes (TP15-17)	15. Remote Set-up
M. 16&22&29	*Spring Break*	*Spring Break*	*Spring Break*	*Spring Break*
Apr. 6	16. Classes, Attributes	VI. Set-up	17. Classes, Methods	18. Special Methods
Apr. 13	19. Classes, OOP	VII. Creating a Class	20. Classes, OOP	21. Classes, OOP
Apr. 20	22. Intro Recursion.	VIII. OOP	23. Recursion II	24. Recursion III
Apr. 27	25. Linked List I	IX. Recursion	26. Linked List II	27. Binary Trees
May 4	28. Iterative Sorting	X. XC Lab	29. Recursive Sorting	30. Search
May 11	31. *Special Topics*	No Lab	32. *Special Topics*	33. Review

Comments from previous renditions

"1. Go to office hours, 2. Go to TA sessions, 3. Don't stress about homeworks."
"Go to office hours! GO TO OFFICE HOURS! Go to office hours!" ⋆ "Read the textbook."
"Look at the code posted after class; don't try to copy it down in class." ⋆ "Think about how your code should works logically before typing anything." ⋆ "Stop complaining and start coding!!!"
"Don't be intimidated...a programming language is just a language...practice the idioms."
"Practice writing code outside of class." ⋆ "Write code on paper beforehand; it helps to pinpoint errors."
"TAs are soooo helpful and just great to talk to." ⋆ "You are learning a lot...Enjoy!" ⋆