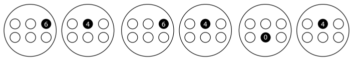
Computer Science 134C: Introduction to Computer Science
Duane A. Bailey
`bailey@cs.williams.edu`

Office:	TPL 306.
Office Hours:	TBA.
Co-instructor:	Iris Howley (`iris@cs.williams.edu`), TCL 308.
Assistants:	Will Burford, Elizabeth Button, Peter Christie, Javier Esparza, Abigail Fournier, Ariel Koltun-Fromm, David Lee, Grace Mazzarella, Dzung Pham, Alex Smith-Bove, Alex Taylor
TA Hours:	TBA.
Text:	Allen Downey's *Think Python, 2ed*, at `greenteapress.com/thinkpython2/thinkpython2.pdf`.
Web resources:	`cs.williams.edu/~bailey/cs134`
Technical Support:	Mary Bailey (`mary@cs.williams.edu`), TCL 312.
Lecture:	Physics 203, Monday, Wednesday, and Friday at 9:00 a.m.
Lab Times:	Mon 1-2:30pm (Duane), 2:30-4pm (Iris), Tue 8:30-10am (I), 10-11:30am (D), 2:30-4pm (I)
Lab Location:	TCL 217a
CS Lab Code:	6-4-6-4-0-4 (remember visually, or **think**: 8^2, 8^2, 2^2).



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 through the building of new tools 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. During formal lab hours, we will meet for 90 minutes to begin work on a more extended problem. We expect that this work will be continued outside of scheduled time. As the end of the semester nears, we'll individualize common projects to focus on topics of our own interest. On occasion, we may meet in lab during our lecture hours.

Work. You are responsible for reading supporting material (*Think Python* (TP)) and participating as the semester progresses. In addition, some topics may require you to investigate online resources (documentation, tutorials, and the like). Each week you will be responsible for completing a programming assignment (35%) in addition to a written homework (15%). **There will be a midterm examination on October 16** (25%), and a scheduled final (T.B.A., 25%). We reserve the right to adjust grades by as much as 5% to reflect course participation.

Week of	**Monday**	Lab	**Wednesday**	**Friday**
Sept. 7	—		—	1. Hello, world! (TP1)
Sept. 10	2. Expressions (TP2)	I. Python and Git	3. Functions (TP3)	4. Conditions (TP5-6)
Sept. 17	5. Abstraction (TP4)	II. Procedure	6. Iteration (TP7)	7. Strings (TP8-9)
Sept. 24	8. Interpretation	III. Toolbox Building	9. Lists, Tuples (TP10,12)	10. Sets, Dicts (TP11)
Oct. 1	11. Files (TP14)	IV. Debugging	12. Generators	*Mountain Day?*
Oct. 10	*Reading Period*	*Reading Period*	13. Iterators	14. Classes (TP15-17)
Oct. 15	15. Special Methods	V. Generators	*Slack*	16. Properties
Oct. 22	17. Classes	VI. Class Design	18. Linked Lists	19. Linked Lists
Oct. 29	20. Beautiful Soup	VII. Visualization	21. Regular Expressions	22. *Slack*
Nov. 5	23. Sorting I	VIII. Trivia	24. Sorting II.	25. Program annotation
Nov. 12	26. Natural Language	IX. Ind. Project	27. Images	28. Binary Trees
Nov. 19	29. Object Persistence	*Break*	*Break*	*Break*
Nov. 26	30. Java I.	X. Java	31. Java II.	32. Java III.
Dec. 3	33. Java IV.	X. Java (cont.)	34. *Slack*	35. Evaluations

Comments from last spring

"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 Duane posts 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!!! Feel swervy!"
"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!" ⋆ "Stay swervy."

Intellectual Property. No part of this course may be reproduced and distributed in any manner without prior permission from the instructors.
Community. We embrace diversity. We welcome all students and expect everyone to contribute and support a respectful and welcoming environment. If have concerns, please share them with us or the college adminstration.
Mental Health. Students who need accommodations should contact the Director of Accessible Education. If you are experiencing health challenges significantly affecting your academic work or well-being, please contact one of us, or a staff member from the Dean's Office.
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.
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`

Anonymous ID. We grade anonymously. When asked, your Anonymous ID (AID) is: