Computer Science 134C: Introduction to Computer Science — Spring 2019
Instructors: Shikha Singh (shikha@cs.williams.edu) and Iris Howley (iris@cs.williams.edu)

Student Help Hours: Shikha (TBL309b): Mon. 2:30-4:00p, Wed. 12:30-2:00p (CS Common Room), Thu. 1-2p
Iris (TCL308): Wed. 12:00-1:00p, Thurs. 10:00a-12:00p
Lida (TCL205) Wed. 1:30-3:00p in the CS Common Room, 3rd floor of TCL (starting March 1)

Assistants: Harun Curak, Diego Esparza, Nathan Thimothe, Emily Zheng, Maria Chapman, Amelia Chen, Caleb Dittmar, Hugo Hua, Brian Kamau, Sarah Lyell, Yash Mangal, Rachel Nguyen, Minh Phan, Mira Sneirson, Jules Walzer-Goldfeld, Emma Wuerth

TA Hours: Sun-Thu 7-9:30p

Course Text: Allen Downey’s Think Python, 2ed, at greenteapress.com/thinkpython2/thinkpython2.pdf.

Technical Support: Lida Doret (lida@cs.williams.edu), TCL 205 & Mary Bailey (mary@cs.williams.edu), T Sl 312.

Lecture: SSL30A, Monday, Wednesday, and Friday at 8:00 a.m (Shikha) & 11:00 a.m (Iris).

Lab Times: Mon 1-2:30pm (Iris), 2:30-4pm (Iris), Tue 1-2:30pm (Shikha), 2:30-4pm (Shikha)

Lab Location: TCL 217a

CS Lab Code: 1-2-4-8-16 (remember visually, or think: 20, 21, 22, 23, 24).

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. 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. There are also weekly written homework assignments to support lecture and lab learning.

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 (40%) in addition to a written
homework (15%). There will be a midterm examination on March 12 (20%), and a scheduled final (T.B.A.,
25%). 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 day 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: To Be Determined.

Attendance Policy. Attendance is not required in the lectures or labs but regular absences may significantly affect
your class participation grade. If you have to miss class due to a conflict, you should inform your instructor. It is
your responsibility to make up for missed work.

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 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](http://csci.williams.edu/the-cs-honor-code-and-computer-usage-policy)

**Lab Grading.** Programming labs will be graded on the following scale:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+</td>
<td>An absolutely fantastic submission of the sort that will only come along a few times during the semester.</td>
</tr>
<tr>
<td>A</td>
<td>A submission that exceeds our standard expectations for the assignment. The program must reflect additional work beyond the requirements or get the job done in a particularly elegant way.</td>
</tr>
<tr>
<td>A-</td>
<td>A submission that satisfies all the requirements for the assignment – a job well done.</td>
</tr>
<tr>
<td>B+</td>
<td>Submission meets the requirements for the assignment, possibly with a few small problems.</td>
</tr>
<tr>
<td>B</td>
<td>A submission that has problems serious enough to fall short of the requirements for the assignment.</td>
</tr>
<tr>
<td>C</td>
<td>A submission that has extremely serious problems, but nonetheless shows some effort &amp; understanding.</td>
</tr>
<tr>
<td>D</td>
<td>A submission that shows little effort and does not represent passing work.</td>
</tr>
</tbody>
</table>

There is some subjectivity to what makes good style, but the basic goal is to make your ideas as clear and easy to follow as possible. Stylistically, we expect to see programs that exhibit the following:

- Meaningful names used in declarations
- Informative comments
- Good and consistent formatting
- Good choice of Python commands.

**Late days.** You are allowed a total of 3 late days over the semester, with at most 2 late days towards any one lab. You must request a late day in advance on the form, here: [http://bit.ly/s20late](http://bit.ly/s20late).
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!” ★

Tentative Schedule of Topics

<table>
<thead>
<tr>
<th>Week of</th>
<th>Monday</th>
<th>Lab</th>
<th>Wednesday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb. 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feb. 10</td>
<td>2. Expressions (TP2)</td>
<td>I. PYTHON AND GITLAB</td>
<td>3. Functions (TP3)</td>
<td>1. Hello, world! (TP1)</td>
</tr>
<tr>
<td>Feb. 17</td>
<td>4. Conditions (TP5-6)</td>
<td>II. PROCEDURE</td>
<td>5. Iteration (TP7)</td>
<td>Winter Carnival</td>
</tr>
<tr>
<td>M. 22&amp;29</td>
<td>Spring Break</td>
<td>Spring Break</td>
<td>Spring Break</td>
<td>18. Slack</td>
</tr>
<tr>
<td>Apr. 6</td>
<td>19. Images</td>
<td>VII. IMAGES</td>
<td>20. Slack</td>
<td>21. Multiple Classes</td>
</tr>
<tr>
<td>Apr. 27</td>
<td>* Slack</td>
<td>IX. RECURSIVE TREES</td>
<td>28. Object Persistence</td>
<td>29. Scope</td>
</tr>
<tr>
<td>May 11</td>
<td>33. Special Topics</td>
<td>X. PROJECT (CONT.)</td>
<td>34. Special Topics</td>
<td>35. Evaluations</td>
</tr>
</tbody>
</table>