Course Syllabus

Syllabus

Introduction to the intellectual enterprises of computer science and to the art of programming. Students learn how to think algorithmically and solve problems efficiently. Topics include abstraction, algorithms, data structures, encapsulation, resource management, security, software engineering, web development, and artificial intelligence. Languages include C, Python, SQL, and JavaScript, plus CSS and HTML. Problem sets inspired by real-world domains of biology, cryptography, finance, forensics, and gaming. See CS50's website, https://cs50.yale.edu (https://cs50.yale.edu), for additional information.

No previous programming experience required. Open to students of all levels and majors.

All lectures were recorded at Harvard in the Fall of 2019; you may watch the recorded lectures any time in advance of section. Section meetings with the instructor will be held Monday and Wednesday from 1pm until 3:15pm.

Expectations

Students are expected to

- watch all lectures (before your section starts),
- attend eight regular sections,
- solve eight problem sets,
- take 5 quizzes,
- design and implement a final project.

Objectives

Educational Objectives

Among the overarching goals of this course, independent of its curriculum, are to

- inspire students to explore unfamiliar waters, without fear of failure;
- create an intensive, shared experience, accessible to all students; and
- build community among students.

Learning Objectives

Among the overarching goals for students individually in this course is that they learn something that we haven't taught them, as is manifest at term's end by so many students' final projects that use languages, libraries, tools, and techniques not taught in the course. Along the way will students learn to

- think more methodically;
• program procedurally;
• represent and process information;
• communicate succinctly and precisely;
• solve problems efficiently;
• recognize patterns among problems;
• decompose problems into parts and compose solutions thereto;
• operate at multiple levels of abstraction;
• separate design from implementation details;
• infer from first principles how systems work;
• assess the correctness, design, and style of code;
• teach themselves new languages;
• identify threats to privacy and security;
• read documentation, drawing conclusions from specifications;
• test solutions to problems, find faults, and identify corner cases;
• describe symptoms of problems precisely and ask questions clearly; and
• identify and quantify tradeoffs among resources, particularly time and space.

Ultimately, the course provides students with a foundation for further studies in computer science and empowers students to apply computer science to problems in other domains.

Curricular Objectives

Week 0 # Scratch

• Problem solving
• Inputs, Outputs
• Representation
• Unary, Binary, Decimal
• Abstraction
• ASCII, Unicode
• RGB
• Algorithms
• Running Time
• Pseudocode
• Scratch
  • Functions, Arguments, Return Values
  • Variables
  • Boolean Expressions, Conditions
  • Loops
  • Events
  • Threads

Week 0 # C
- Linux
- Command-Line Interface
- C
  - Functions, Arguments, Return Values
  - Variables
  - Boolean Expressions, Conditions
  - Loops
- Libraries, Header Files
- Text Editors
- Terminal Windows
- Compiler
- Types
- Integer Overflow
- Floating-Point Imprecision

**Week 1 # Arrays**
- Preprocessing
- Compiling
- Assembling
- Linking
- Debugging
- Arrays
- Strings
- Command-Line Arguments
- Cryptography

**Week 1 # Algorithms**
- Searching
  - Linear Search
  - Binary Search
- Sorting
  - Bubble Sort
  - Selection Sort
  - Insertion Sort
  - Merge Sort
- Asymptotic Notation
  - $O$
  - $\Omega$
  - $\Theta$
- Recursion
Week 2 # Python

- Python
  - Functions, Arguments, Return Values
  - Variables
  - Boolean Expressions, Conditions
  - Loops
- Modules, Packages

Week 2 # SQL

- SQL
  - Tables
  - Types
  - Statements
  - Constraints
  - Indexes
  - Keywords, Functions
  - Transactions
- Race Conditions
- SQL Injection Attacks

Week 3 # Web

- Python Flask Framework
  - HTML
  - CSS
  - Javascript
- SQL Database Integration

Grades

You are encouraged to take CS50 Credit/D/Fail if you will feel less nervous without the pressure of a letter grade. Note it will only satisfy the QR requirement if you take it for a grade. Please also remember that your course grade will reflect how much you progress over the semester. We expect you to work hard and learn a lot, and your course grades will reflect that. Course grades tend to be quite high in CS50.

Whether taking the course Credit/D/Fail or for a letter grade, you must ordinarily submit all problem sets, brief pre-section quizzes, one exam, and submit a final project unless granted an exception in writing by the course's heads. Multiple missing problem sets, a missing quiz or final project, and violations of the academic honesty policy may each result in a reduced or failing grade.

Graduate and professional students are expected to produce a final project that is 50% greater in scope than what is expected of undergraduate students. Your course grade will be adapted to the prevailing
standards of your school, but we encourage you to take the class Pass/Fail if your degree program permits it.

Auditors are welcome in CS50. Graduate students, employees, and their family members who formally register as auditors may also participate in all aspects of the class including section, office hours, and receiving graded feedback on your problem sets. We do not require or expect you to complete all the coursework like a regular student, and we will not worry if you choose not to. However, we will not be able to help you individually catch up or grade late submissions if you have not been fully participating in the class.

Final grades are determined using the following weights:

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem Sets</td>
<td>50%</td>
</tr>
<tr>
<td>Quizzes</td>
<td>10%</td>
</tr>
<tr>
<td>Final Project</td>
<td>20%</td>
</tr>
<tr>
<td>Attendance*</td>
<td>10%</td>
</tr>
<tr>
<td>Participation</td>
<td>10%</td>
</tr>
</tbody>
</table>

* At sections.

Know that CS50 draws quite the spectrum of students, including "those less comfortable," "those more comfortable," and those somewhere in between. However, what ultimately matters in this course is not so much where you end up relative to your classmates but where you end up relative to yourself at the start.

Each student's final grade is individually determined at term's end. Remarkable effort and upward trending are considered, as is input from the teaching fellows. The course does not have pre-determined cutoffs for final grades. The course is not graded on a curve. Those less comfortable and somewhere in between are not at a disadvantage vis-à-vis those more comfortable.

**Books**

No books are required or recommended for this course. However, you might find the below books of interest. Realize that free, if not superior, resources can be found on the course's website.

Lectures will be available to watch on demand.

<table>
<thead>
<tr>
<th>#</th>
<th>Lecture</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>C</td>
<td>Mon 6/7</td>
</tr>
<tr>
<td>1</td>
<td>Arrays</td>
<td>Wed 6/9</td>
</tr>
<tr>
<td>2</td>
<td>Algorithms</td>
<td>Mon 6/14</td>
</tr>
<tr>
<td>3</td>
<td>Memory</td>
<td>Wed 6/16</td>
</tr>
<tr>
<td>4</td>
<td>Python</td>
<td>Mon 6/21</td>
</tr>
<tr>
<td>5</td>
<td>SQL</td>
<td>Wed 6/23</td>
</tr>
<tr>
<td>6</td>
<td>Web: HTML, CSS, JavaScript</td>
<td>Mon 6/28</td>
</tr>
<tr>
<td>7</td>
<td>Web: Flask</td>
<td>Wed 6/30</td>
</tr>
</tbody>
</table>

Sections

Lectures are supplemented by sections led by the instructor, held during the scheduled class time from 1pm - 3:15pm each Monday and Wednesday. Attendance at section is required.

Office Hours

Office hours are opportunities for help with problem sets alongside the instructor. An open-format office hours will be held during the scheduled class time from 1pm - 3:15pm each Friday. Small group and individual office hours will be available by appointment.

Problem Sets

Problem sets will be assigned at a rate of 2 per week. Late work is not ordinarily accepted, except with a Dean's excuse. (Graduate students should e-mail heads@cs50.yale.edu as early as possible to request an extension.)
<table>
<thead>
<tr>
<th>#</th>
<th>Language</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>C</td>
<td>Sun 6/13, 11:59pm</td>
</tr>
<tr>
<td>1</td>
<td>C</td>
<td>Sun 6/13, 11:59pm</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>Sun 6/20, 11:59pm</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>Sun 6/20, 11:59pm</td>
</tr>
<tr>
<td>4</td>
<td>Python</td>
<td>Sun 6/27, 11:59pm</td>
</tr>
<tr>
<td>5</td>
<td>SQL</td>
<td>Sun 6/27, 11:59pm</td>
</tr>
<tr>
<td>6</td>
<td>Web</td>
<td>Fri 7/2, 11:59pm</td>
</tr>
</tbody>
</table>

**Quizzes**

Quizzes are short assignments due after each lecture that allow you to apply each week's concepts to new problems. Each quiz is open-book: you may use any and all non-human resources during a quiz, but the only humans to whom you may turn for help or from whom you may receive help are the course's heads. Your 5 highest quiz scores will be counted towards your final grade. **All quizzes are due before your section starts!**

<table>
<thead>
<tr>
<th>#</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Wed 6/9</td>
</tr>
<tr>
<td>1</td>
<td>Mon 6/14</td>
</tr>
<tr>
<td>2</td>
<td>Wed 6/16</td>
</tr>
<tr>
<td>3</td>
<td>Mon 6/21</td>
</tr>
<tr>
<td>4</td>
<td>Wed 6/23</td>
</tr>
</tbody>
</table>

**Final Project**
The climax of this course is its final project. The final project is your opportunity to take your newfound savvy with programming out for a spin and develop your very own piece of software. So long as your project draws upon this course's lessons, the nature of your project is entirely up to you, albeit subject to the staff's approval. You may implement your project in any language(s) as long as the staff approves. You are welcome to utilize any infrastructure, provided you have access to any hardware and software that your project requires. All that we ask is that you build something of interest to you, that you solve an actual problem, that you impact campus, or that you change the world! Strive to create something that outlives this course.

Inasmuch as software development is rarely a one-person effort, you are allowed an opportunity to collaborate with one or two classmates for this final project. Needless to say, it is expected that every student in any such group contribute equally to the design and implementation of that group's project. Moreover, it is expected that the scope of a two- or three-person group's project be, respectively, twice or thrice that of a typical one-person project. A one-person project, mind you, should entail more time and effort than is required by each of the course's problem sets. Although no more than three students may design and implement a given project, you are welcome to solicit advice from others, so long as you respect the course's policy on academic honesty.

Extensions on the final project always require a Dean's excuse (graduate students must provide documentation of a genuine emergency). Late submissions will receive no credit.

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposal</td>
<td>Sun 7/4, 11:59pm</td>
</tr>
<tr>
<td>Status Report</td>
<td>Tue 7/6, 11:59pm</td>
</tr>
<tr>
<td>Implementation</td>
<td>Thu 7/8, 11:59pm</td>
</tr>
<tr>
<td>Peer Feedback</td>
<td>Fri 7/9, 11:59pm</td>
</tr>
</tbody>
</table>

**CS50 (Virtual) Fair**

From 1pm until 3pm on Fri 7/9 via Zoom, an epic display of final projects. Not only is the CS50 Fair a chance to see classmates' projects and demo your own, it is an opportunity to mingle with students, faculty, and staff from across campus as well as recruiters from industry. Attendance is expected of all students.

**Financial Aid**

CS50 does not require that students purchase any books or software. Access to a computer with internet access and ability to video-conference via Zoom is required. Students will not have access to
on-campus computers, and equipment rentals from Yale libraries will not be offered for the summer term.

**Academic Honesty**

This course's philosophy on academic honesty is best stated as "be reasonable." The course recognizes that interactions with classmates and others can facilitate mastery of the course's material. However, there remains a line between enlisting the help of another and submitting the work of another. This policy characterizes both sides of that line.

The essence of all work that you submit to this course must be your own and you must explicitly cite anyone you collaborate with and any resources you use that are not part of the course material or directly linked from the pset instructions.

- You may ask classmates and others for help on psets, and you may use outside resources that do not reduce to another doing your work for you. You must include comments in your code with the full name of the person you discussed with ("John Q. Adams" is good, "my roommate," "my tutor," or "Natalie" is not), or the complete URL or reference for the outside resource; what you discussed, and how that helped you. This is partly a note-taking process and partly a matter of properly citing sources.
- You must also cite any help or discussions you have with other students in the class. If the pset doesn't explicitly include a place for this, do so in comments in the most related section of your own code.
- Generally speaking, when asking for help, you may show your code to others, but you may not view theirs. You may not post your code online, and you may not look at or use online solutions to the psets.
- Collaboration on the course's quiz is not permitted at all. Collaboration on the course's final project is permitted to the extent prescribed by its specification.

**Regret clause.** If you commit some act that is not reasonable but bring it to the attention of the course's heads within 72 hours, the course may impose local sanctions that may include an unsatisfactory or failing grade for work submitted, but the course will not refer the matter for further disciplinary action unless another infraction occurs. You may invoke this clause only once for a true infraction, but if you self-report something that the course heads feel is not a significant violation, it will not count against you or count as your one invocation.

In cases of suspected violations involving students at both Harvard and Yale, students will be referred to the appropriate committee at their university. Those committees may exchange information for the purpose of resolving the cases in accordance with their own procedures. They may also reach different conclusions and impose different sanctions from the same set of facts and evidence.

Below are rules of thumb that (inexhaustively) characterize acts that the course considers reasonable and not reasonable. If in doubt as to whether some act is reasonable, do not commit it until you solicit and receive approval in writing from the course's heads. Acts considered not reasonable by the course
are handled harshly. If the course refers some matter for disciplinary action and the outcome is punitive, the course reserves the right to impose local sanctions on top of that outcome that may include an unsatisfactory or failing grade for work submitted or for the course itself.

**Reasonable**

- Communicating with classmates about problem sets' problems in English (or some other spoken language), and properly citing those discussions.
- Discussing the course's material with others in order to understand it better. You do not need to cite this if it isn't related to the pset, but we encourage citing and note-taking on your discussion anyway.
- Helping a classmate identify a bug in his or her code at office hours, elsewhere, or even online, as by viewing, compiling, or running his or her code after you have submitted that portion of the pset yourself. Add a citation to your own code of the help you provided and resubmit.
- Incorporating a few lines of code that you find online or elsewhere into your own code, provided that those lines are not themselves solutions to assigned problems and that you cite the lines' origins.
- Reviewing past semesters' tests and quizzes and solutions thereto.
- Sending or showing code that you've written to someone, possibly a classmate, so that he or she might help you identify and fix a bug, provided you properly cite the help. If it is a classmate, make sure they cite giving the help as well.
- Submitting the same or similar work to this course that you have submitted previously to this course, CS50 AP, or CS50x. Please include a comment at the top of the file indicating it is a resubmission to eliminate any potential confusion.
- Turning to the course's heads for help or receiving help from the course's heads during the quiz.
- Turning to the web or elsewhere for instruction beyond the course's own, for references, and for solutions to technical difficulties, but not for outright solutions to problem set's problems or your own final project. Cite any such resources, especially if they are related to a pset.
- Whiteboarding solutions to problem sets with others using diagrams or pseudocode but not actual code. Cite who you discussed with and what you discussed, even if you are doing this in office hours under TA supervision. It helps avoid misunderstandings down the road.
- Working with (and even paying) a tutor to help you with the course, provided the tutor does not do your work for you and you cite the pset help you receive.

**Not Reasonable**

- Accessing a solution to some problem prior to (re-)submitting your own.
- Accessing or attempting to access, without permission, an account not your own.
- Asking a classmate to see his or her solution to a problem set's problem before (re-)submitting your own.
- Decompiling, deobfuscating, or disassembling the staff's solutions to problem sets.
- Discovering but failing to disclose to the course's heads bugs in the course's software that affect scores.
• Failing to cite (as with comments) the origins of code or techniques that you discover outside of the course’s own lessons and integrate into your own work, even while respecting this policy’s other constraints.
• Giving or showing to a classmate a solution to a problem set’s problem when it is he or she, and not you, who is struggling to solve it.
• Looking at another individual’s work during the quiz.
• Manipulating or attempting to manipulate scores artificially, as by exploiting bugs or formulas in the course’s software.
• Paying or offering to pay an individual for work that you may submit as (part of) your own.
• Providing or making available solutions to problem sets to individuals who might take this course in the future.
• Searching for or soliciting outright solutions to problem sets online or elsewhere.
• Splitting a problem set’s workload with another individual and combining your work.
• Submitting (after possibly modifying) the work of another individual beyond the few lines allowed herein.
• Submitting the same or similar work to this course that you have submitted or will submit to another.
• Submitting work to this course that you intend to use outside of the course (e.g., for a job) without prior approval from the course’s heads.
• Turning to humans (besides the course’s heads) for help or receiving help from humans (besides the course’s heads) during the quiz.
• Viewing another’s solution to a problem set’s problem and basing your own solution on it.

Acknowledgement and Authorization

Harvard plans to record audio, photos, and video of Computer Science 50 (CS50) lectures, sections, office hours, seminars, and other events and activities related to CS50 (the "Recordings"), with the aims of making the content of the course more widely available and contributing to public understanding of innovative learning (the "Projects"). The Recordings, or edited versions of them, may be made available to other Harvard students, to students at other educational institutions, and to the broader public via edX, the Internet, television, theatrical distribution, digital media, or other means. It is also possible that the Recordings may be used to make other derivative works in the future. Students may elect not to appear in photos and video used in the Projects and may still participate fully in CS50.

When you submit Problem Set 0, you will need to sign online an Acknowledgement and Authorization in the following form:

I understand that, if I do not wish any photos or video of me to be used as part of the Projects, I should so inform the course's instructor by emailing recordings@cs50.harvard.edu (mailto:recordings@cs50.harvard.edu), within one week of enrolling in CS50. In that event, I understand that I should sit in the designated "no-film" zone of CS50 classrooms and should not walk in the field of view of the cameras. I understand that Harvard will take reasonable steps, with my cooperation, to avoid including identifiable images of me in the Projects' photos and video shot in classrooms and other
course locations after I opt out as just described. I understand that I am free to opt out of the Projects' photos and video in this way, and that doing so will not affect my grade or my ability to participate in course activities.

Unless I opt out of the Projects' photos and video as described above and take the steps that will be outlined by the instructor to avoid being filmed, I authorize Harvard and its designees to record and use photos and video of my participation in CS50 and activities related to CS50 (the "Recordings"). I understand and agree that the Recordings may include my image, name, and voice. I also understand and agree that, even if I opt out of the Projects' photos and video, my spoken name and voice may be picked up by microphones outside the "no-film" zone and may be included in the Recordings.

I understand and agree that Harvard and its designees will have the irrevocable, worldwide right to make, edit, modify, copy, publish, transmit, distribute, sell, publicly display, publicly perform, and otherwise use and make available its respective Recordings and any other works that may be derived from those Recordings, in any manner or medium now known or later invented, and to authorize others to do so as well. I hereby transfer to Harvard any rights, including copyrights, I may have in the Recordings that Harvard makes. I will remain free to use and disseminate any ideas, remarks, or other material that I may contribute to course discussions.

I acknowledge and agree that I will not be entitled to any payment, now or in the future, in connection with the Recordings or any works derived from them. This Acknowledgment and Authorization is a binding agreement, and is signed as a document under seal governed by the laws of the Commonwealth of Massachusetts.

Unless you opt out as described in the Acknowledgment and Authorization, you are agreeing, by attending CS50, that your participation in CS50 and related activities may be recorded and used by Harvard in connection with the Projects without further obligation or liability to you, even if you do not sign any authorization.

If you have any questions about the above, contact recordings@cs50.harvard.edu (mailto:recordings@cs50.harvard.edu).

### Course Summary:

<table>
<thead>
<tr>
<th>Date</th>
<th>Details</th>
<th>Due</th>
</tr>
</thead>
</table>

https://yale.instructure.com/courses/67332/assignments/syllabus