
Syllabus

This is CS50. Harvard University. Fall 2014.

Table of Contents

Description	2
Expectations	3
Website	3
Grades	3
Books	4
For Those Less Comfortable	4
For Those More Comfortable	4
Lectures	5
Week 0	5
Week 1	6
Week 2	6
Week 3	6
Week 4	6
Week 5	6
Week 6	6
Week 7	7
Week 8	7
Week 9	7
Week 10	7
Week 11	7
Week 12	7
Sections	8
Office Hours	8
Walkthroughs	8
Postmortems	8
Tutoring	9
Problem Sets	9
Problem Set 0	10
Problem Set 1	10

Problem Set 2	10
Problem Set 3	10
Problem Set 4	10
Problem Set 5	10
Problem Set 6	10
Problem Set 7	10
Problem Set 8	10
Quizzes	11
Quiz 0	11
Quiz 1	11
Final Project	11
Pre-Proposal	12
Proposal	12
Status Report	12
CS50 Hackathon	12
Implementation	12
CS50 Fair	13
CS50 Hackathon	13
CS50 Fair	13
Academic Honesty	13
Reasonable	14
Not Reasonable	15
Acknowledgement and Authorization	16

version 0

Description

Introduction to the intellectual enterprises of computer science and the art of programming. This course teaches students how to think algorithmically and solve problems efficiently. Topics include abstraction, algorithms, data structures, encapsulation, resource management, security, software engineering, and web development. Languages include C, PHP, and JavaScript plus SQL, CSS, and HTML. Problem sets inspired by real-world domains of biology, cryptography, finance, forensics, and gaming. Designed for concentrators and non-concentrators alike, with or without prior programming experience

Expectations

You are expected to submit nine problem sets, take two quizzes, and submit a final project.

Website

This course lives at <https://cs50.harvard.edu/>.

Visit the course's website to watch videos, to get help, to download handouts and software, and to follow links to other resources.

Grades

All students must ordinarily submit all nine problem sets, take both quizzes, and submit a final project in order to be eligible for a satisfactory grade unless granted an exception in writing by the course's heads.

Final grades are determined using the following weights:

Problem Sets	50%
Quizzes	40%
Final Project	10%

Problem sets and the final project are evaluated primarily along axes of scope, correctness, design, and style, with scores ordinarily determined by $\text{scope} \times (3 \times \text{correctness} + 2 \times \text{design} + 1 \times \text{style})$. Scores are normalized across TFs at term's end, so mid-semester comparisons among students of scores are not reliable indicators of standing.

Although you must submit all nine problem sets, your lowest score among those problem sets on which you received perfect scores for the axis of scope will be dropped when final grades are determined.

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, in Week 12, end up relative to yourself in Week 0.

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

Books

No books are required for this course. However, you may want to supplement your preparation for or review of some lectures with self-assigned readings relevant to those lectures' content from either of the books below. The first is intended for those inexperienced in (or less comfortable with the idea of) programming. The second is intended for those experienced in (or more comfortable with the idea of) programming. Both are available at the Coop and at sites like [Amazon.com](http://www.amazon.com/)¹. Both have been placed on reserve at [Cabot Library](http://hcl.harvard.edu/libraries/cabot/)². Realize that free, if not superior, resources can be found on the [course's website](https://cs50.harvard.edu/resources)³.

For Those Less Comfortable

C Programming Absolute Beginner's Guide, Third Edition
Greg Perry, Dean Miller
Pearson Education, 2014
ISBN 0-789-75198-4

For Those More Comfortable

Programming in C, Fourth Edition
Stephen G. Kochan
Pearson Education, 2015
ISBN 0-321-77641-0

¹ <http://www.amazon.com/>

² <http://hcl.harvard.edu/libraries/cabot/>

³ <https://cs50.harvard.edu/resources>

The book below is recommended for those interested in understanding how their own computers work for personal edification. It is available at the Coop and at sites like [Amazon.com](http://www.amazon.com/)⁴. And it, too, has been placed on reserve.

How Computers Work, Tenth Edition

Ron White

Que Publishing, 2014

ISBN 0-7897-4984-X

This last book below is recommended for aspiring hackers, those interested in programming techniques and low-level optimization of code for applications beyond the scope of this course. It is also available at the Coop and at sites like [Amazon.com](http://www.amazon.com/)⁵. And it, too, has been placed on reserve.

Hacker's Delight, Second Edition

Henry S. Warren Jr.

Pearson Education, 2013

ISBN 0-321-84268-5

Lectures

Most lectures take place at Harvard in Sanders Theatre on Mondays and Wednesdays from 1pm until 2pm. (Four lectures will take place at Yale.) All lectures are live-streamed online and are available on demand immediately thereafter. You are welcome to watch lecture online if you are unable to attend some lecture in person (particularly at Yale!). You are encouraged to watch lectures online for the sake of review.

A schedule of lectures, subject to change, appears below.

Week 0

Wed 9/2, Fri 9/4

Binary. ASCII. Algorithms. Pseudocode. Source code. Compiler. Object code. Scratch. Statements. Boolean expressions. Conditions. Loops. Variables. Functions. Arrays. Threads. Events.

⁴ <http://www.amazon.com/>

⁵ <http://www.amazon.com/>

Week 1

Wed 9/9, Fri 9/11

Linux. C. Compiling. Libraries. Types. Standard output.

Week 2

Mon 9/14, Wed 9/16

Casting. Imprecision. Switches. Scope. Strings. Arrays. Cryptography.

Week 3

Mon 9/21, Wed 9/23

Command-line arguments. Searching. Sorting. Bubble sort. Selection sort. Insertion sort. O . Ω . Θ . Recursion. Merge Sort.

Week 4

Mon 9/28, Wed 9/30

Stack. Debugging. File I/O. Hexadecimal. Strings. Pointers. Dynamic memory allocation.

Week 5

Mon 10/5, Wed 10/7

Heap. Buffer overflow. Linked lists.

Week 6

Mon 10/12, Wed 10/14, Fri 10/16

Hash tables. Tries.

Quiz 0.

Trees. Stacks. Queues.

Week 7

Mon 10/19, Wed 10/21

TCP/IP. HTTP. HTML. CSS.

Week 8

Mon 10/26, Wed 10/28

PHP. MVC. SQL.

Week 9

Mon 11/2, Wed 11/4

JavaScript. Ajax.

Week 10

Mon 11/9, Wed 11/11

Security. Artificial intelligence.

Week 11

Mon 11/16, Wed 11/18

Artificial intelligence, continued.

Quiz 1.

Week 12

Mon 11/23

Exciting conclusion.

Sections

Lectures are supplemented by weekly, 90-minute sections led by the teaching fellows. Sections provide you with opportunities to explore the course's material in a more intimate environment, with only your teaching fellow and a handful of classmates present, as well as to dive into hands-on activities.

Different sections are offered for those less comfortable, those more comfortable, and those somewhere in between.

Sectioning begins in Week 1. Sections themselves begin in Week 3, with course-wide supersections (open to all students) offered in Week 2.

A schedule of sections (and supersections) will appear on the course's website.

Office Hours

Office hours are opportunities for assistance with problem sets alongside the course's teaching fellows and course assistants. Office hours ordinarily take place from 9pm until midnight in Widener Library on Mondays and Tuesdays and in Annenberg Hall on Wednesdays and Thursdays.

Upon arrival, you'll be greeted by the course's staff, who'll refer you to a particular table for help.

Walkthroughs

Integrated into problem sets are "walkthroughs," videos via which the course's staff offer direction on where to begin and how to approach a challenge. You are expected to watch walkthroughs before asking questions about problem sets at office hours or via [CS50 Discuss](#)⁶.

Postmortems

Available after problem sets' deadlines are "postmortems," videos via which the course's staff explore actual solutions to problem sets. You are expected to watch postmortems for insights into how else you could have (or should have!) implemented your own solutions.

⁶ <https://cs50.harvard.edu/discuss>

Tutoring

For students who avail themselves of the course's other resources but still find themselves struggling, the course offers tutoring, to the extent possible, providing opportunities for students to work with the course's staff one-on-one or in particularly small groups. To arrange, contact the course's heads.

Problem Sets

Nine problem sets are assigned during the semester. Each is due by noon on a Thursday. However, you have nine "late days" that you may "spend" during the semester, each of which provides you with an extension of twenty-four hours. **You may spend no more than one late day on any particular problem set.** You need not inform the staff of your use of a late day; usage of late days is tracked automatically. Lateness of electronic submissions is determined down to the minute by submissions' timestamps. Submitting more than seven minutes late is equivalent to submitting twenty-four hours late. Late work is not ordinarily accepted once you have exhausted your late days, except in cases of medical emergency. These late days cannot be spent on the course's final project.

In order to accommodate students with different backgrounds, some problem sets are released in two editions: a standard edition intended for most students and a "Hacker Edition" intended for some students. Both editions essentially cover the same material. But the Hacker Edition typically presents that material from a more technical angle and poses more sophisticated questions. Hacker Editions are graded separately from standard editions, but those students who submit the former do not receive any form of extra credit outright. When determining grades at term's end, however, we do bear in mind submissions of Hacker Editions.

To be clear, we encourage most students (including aspiring computer scientists) to tackle the standard editions. However, you may choose, week to week, which edition to submit. You may not submit both or some amalgam of the two.

Although you must submit all nine problem sets, your lowest score among those problem sets on which you received a perfect score for the axis of scope will be dropped when final grades are determined.

A schedule of problem sets, subject to change, appears below.

Problem Set 0

due by noon on Thu 9/10

Problem Set 1

due by noon on Thu 9/17

Problem Set 2

due by noon on Thu 9/24

Problem Set 3

due by noon on Thu 10/1

Problem Set 4

due by noon on Thu 10/8

Problem Set 5

due by noon on Thu 10/22

Problem Set 6

due by noon on Thu 10/29

Problem Set 7

due by noon on Thu 11/5

Problem Set 8

due by noon on Thu 11/12

Quizzes

The course has two 75-minute quizzes. These quizzes are "closed-book," but you may utilize during each quiz one two-sided page (8.5" × 11") of handwritten or typed notes, blank scrap paper, and a pen or pencil, nothing else.

When final grades are computed, your scores on these two quizzes are weighted equally.

Unless arranged with the course's heads in advance, quizzes cannot ordinarily be taken at alternative times even if missed by accident, except in cases of medical emergency.

A schedule of quizzes, subject to change, appears below. These quizzes take place at 1pm in lieu of lectures on these dates. Simultaneously enrolled students are expected to sit for these quizzes at 5:30pm on these same dates unless otherwise arranged with the course's heads in advance.

Quiz 0

Wed 10/14

Covers weeks 0 through 5.

Quiz 1

Wed 11/18

Covers weeks 0 through 11 with emphasis on 6 onward.

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 the staff ultimately has 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 are not ordinarily granted, except in cases of medical emergency. Technical difficulties are not considered emergencies. Problem sets' late days cannot be spent on the final project. Lateness of submissions is determined down to the minute by submissions' timestamps. Submitting more than seven minutes late is equivalent to not submitting at all.

Pre-Proposal

due by noon on Mon 11/2

Proposal

due by noon on Mon 11/9

Status Report

due by noon on Mon 11/30

CS50 Hackathon

from 7pm on Thu 12/3 until 7am on Fri 12/4

Implementation

due by noon on Sun 12/6

CS50 Fair

from 11am until 4:30pm on Mon 12/7

CS50 Hackathon

From 7pm on Thu 12/3 until 7am on Fri 12/4 is the CS50 Hackathon, an epic all-nighter during which you can dive into your final project's implementation alongside classmates (from Harvard and Yale alike!) and staff. If you choose to partake, you'll be asked to propose three milestones for yourself that evening: a "good" one that you intend to achieve no matter what; a "better" one that you think you can achieve; and a "best" one that you hope to achieve.

Dinner will be served around 9pm, second dinner will be served around 1am, and those still standing around 5am will be treated to breakfast at IHOP.

CS50 Fair

From 11am until 4:30pm on Mon 12/7 is the CS50 Fair, an epic display of final projects. Not only is the CS50 Fair a venue at which 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.

Also in attendance are popcorn, candy, and a raffle with (fabulous) prizes. Family and friends are welcome to join.

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. Collaboration on problem sets is not permitted except to the extent that you may ask classmates and others for help so long as that help does not reduce to another doing your work for you. Generally speaking, when asking for help, you may show your code to others, but you may not view theirs, so long as you and they respect this policy's other constraints. Collaboration on

quizzes is not permitted at all. Collaboration on the course's final project is permitted to the extent prescribed by its specification.

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 to the Administrative Board and the outcome is Admonish, Probation, Requirement to Withdraw, or Recommendation to Dismiss, 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.

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 to the Administrative Board except in cases of repeated acts.

Reasonable

- Communicating with classmates about problem sets' problems in English (or some other spoken language).
- Discussing the course's material with others in order to understand it better.
- 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, even on your own computer.
- Incorporating snippets of code that you find online or elsewhere into your own code, provided that those snippets are not themselves solutions to assigned problems and that you cite the snippets' origins.
- Reviewing past semesters' 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.
- Sharing snippets of your own code online so that others might help you identify and fix a bug.
- 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.

- Whiteboarding solutions to problem sets with others using diagrams or pseudocode but not actual code.
- Working with (and even paying) a tutor to help you with the course, provided the tutor does not do your work for you.

Not Reasonable

- Accessing a solution to some problem prior to (re-)submitting 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.
- 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 a quiz.
- 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, soliciting, or viewing a quiz's questions or answers prior to taking the quiz.
- 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 allowed snippets.
- 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.

- Using resources during a quiz beyond those explicitly allowed in the quiz's instructions.
- 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.edu⁷ 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 to record and use photos and video of my participation

⁷ <mailto:recordings@cs50.harvard.edu>

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 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.edu⁸.

⁸ <mailto:recordings@cs50.harvard.edu>