

# This is CS50x

OpenCourseWare

Donate 

David J. Malan

malan@harvard.edu



Week 0 Scratch 

Week 1 C

Week 2 Arrays

Week 3 Algorithms

Week 4 Memory

Week 5 Data Structures

Week 6 Python 

Week 7 SQL

Week 8 Information

Tracks

Android

Games

iOS

Web

Final Project

Academic Honesty

## Syllabus

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, and software engineering. Languages include C, Python, and SQL plus students' choice of: HTML, CSS, and JavaScript (for web development); Java or Swift (for mobile app development); or Lua (for game development). Problem sets inspired by the arts, humanities, social sciences, and sciences. Course culminates in a final project. Designed for concentrators and non-concentrators alike, with or without prior programming experience. Two thirds of CS50 students have never taken CS before. Among the overarching goals of this course 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.

## Expectations

You are expected to

- submit nine problem sets and
- submit a final project.

## Certificates

CS50x is free to take, and you are welcome to submit the course's nine problem sets and final project for automated feedback. To be eligible for a [verified certificate](#) from edX, however, you must receive a satisfactory score (at least 70%) on each problem you submit as part of one of the course's nine problem sets as well as on the course's final project.

Problems are evaluated along axes of correctness (as determined by a program called `check50`) and style (as determined by a program called `style50`), with scores ordinarily computed as  $3 \times \text{correctness} + 1 \times \text{style}$ .

## 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.

*Hacker's Delight*, Second Edition

Henry S. Warren Jr.

Pearson Education, 2013

ISBN 0-321-84268-5

*How Computers Work*, Tenth Edition

Ron White

Que Publishing, 2014

ISBN 0-7897-4984-X

*Programming in C*, Fourth Edition

Stephen G. Kochan

Pearson Education, 2015

ISBN 0-321-77641-0

## Lectures

The course's lectures introduce each week's concepts.

## Walkthroughs

Integrated into problem sets are "walkthroughs," videos that offer direction on where to begin and how to approach problems.

## Problem Sets

Problem sets are programming assignments. CS50x does not have deadlines for problem sets. You are welcome to work on and submit them at your own pace. To be eligible for a verified certificate from edX, however, you must submit (and receive a score of at least 70% on) all problem sets by 31 December 2020.

## 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. You may implement your project in any language(s). You are welcome to utilize infrastructure other than the CS50 IDE. All that we ask is that you build something of interest to you, that you solve an actual problem, that you impact your community, 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.

CS50x does not have a deadline for the final project. You are welcome to work on and submit it at your own pace. To be eligible for a verified certificate from

## 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 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. If the course determines that you have committed an act that is not reasonable, you may be deemed ineligible for a certificate. If you commit some act that is not reasonable but bring it to the attention of the course's instructor within 72 hours, the course may reconsider that outcome.

### 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 in person or online, as by viewing, compiling, or running his or her code, even on your own computer.
- 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.
- 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 a few lines 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.
- Paying or offering to pay an individual for work that you may submit as (part of) your own.
- 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.
- Viewing another's solution to a problem set's problem and basing your own solution on it.