This is CS50

CS50's Introduction to Computer Science

OpenCourseWare

Donate ⬈ (https://cs50.harvard.edu/donate)

David J. Malan (https://cs.harvard.edu/malan/)
malan@harvard.edu
𝐟 (https://www.facebook.com/dmalan) ⓞ (https://github.com/dmalan) ⓘ (https://www.instagram.com/davidjmalan/) 𝐢𝐧 (https://www.linkedin.com/in/malan/) ⓡ (https://www.reddit.com/user/davidjmalan) ⓔ (https://www.threads.net/@davidjmalan) 🐦 (https://twitter.com/davidjmalan)

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 Codespace. 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.

> Note that CS50's staff audits submissions to CS50x including this final project. Students found to be in violation of **the Academic Honesty policy** will be removed from the course and deemed ineligible for a certificate. Students who have already completed CS50x, if found to be in violation, will have their CS50 Certificate (and edX Certificate, if applicable) revoked.

Ideas

- a web-based application using JavaScript, Python, and SQL
- an iOS app using Swift
- a game using Lua with LÖVE
- an Android app using Java
- a Chrome extension using JavaScript
- a command-line program using C
- a hardware-based application for which you program some device
- ...

Getting Started

Creating an entire project may seem daunting. Here are some questions that you should think about as you start:

- What will your software do? What features will it have? How will it be executed?
- What new skills will you need to acquire? What topics will you need to research?
- If working with one or two classmates, who will do what?
- In the world of software, most everything takes longer to implement than you expect. And so it's not uncommon to accomplish less in a fixed amount of time than you hope. What might you consider to be a good outcome for your project? A better outcome? The best outcome?

Consider making goal milestones to keep you on track.

If using the CS50 Codespace, create a directory called `project` to store your project source code and other files. You are welcome to develop your project outside of the CS50 Codespace.

> For your final project (and your final project only!) it is reasonable to use AI-based software other than CS50's own (e.g., ChatGPT, GitHub Copilot, Bing Chat, et al.), but the essence of the work must still be your own. You've learned enough to use such tools as

helpers. Treat such tools as amplifying, not supplanting, your productivity. But you still must cite any use of such tools in the comments of your code.

How to Submit

You must complete all three steps!

Step 1 of 3

Create a short video (that's no more than 3 minutes in length) in which you present your project to the world. Your video **must** begin with an opening section that displays:

- your project's title;
- your name;
- your GitHub and edX usernames;
- your city and country;
- and, the date you have recorded this video.

It should then go on to demonstrate your project in action, as with slides, screenshots, voiceover, and/or live action. See howtogeek.com/205742/how-to-record-your-windows-mac-linux-android-or-ios-screen (https://www.howtogeek.com/205742/how-to-record-your-windows-mac-linux-android-or-ios-screen/) for tips on how to make a "screencast," though you're welcome to use an actual camera. Upload your video to YouTube (or, if blocked in your country, a similar site) and take note of its URL; it's fine to flag it as "unlisted," but don't flag it as "private."

Submit this form (https://forms.cs50.io/d5009db5-ee7d-43f1-8214-87cebc1a554f)!

Step 2 of 3

Create a `README.md` text file (named exactly that!) in your `project` folder that explains your project. This file should include your Project Title, the URL of your video (created in step 1 above) and a description of your project. You may use the below as a template.

```
# YOUR PROJECT TITLE
#### Video Demo:  <URL HERE>
#### Description:
TODO
```

If unfamiliar with Markdown syntax, you might find GitHub's Basic Writing and Formatting Syntax (https://docs.github.com/en/free-pro-team@latest/github/writing-on-github/basic-

writing-and-formatting-syntax) helpful. You can also preview your `.md` file by clicking the 'preview' icon as explained here: Markdown Preview in vscode (https://code.visualstudio.com/docs/languages/markdown#_markdown-preview). Standard software project `README`s can often run into the thousands or tens of thousands of words in length; yours need not be that long, but should at least be several hundred words that describe things in detail!

> Your `README.md` file should be minimally multiple paragraphs in length, and should explain what your project is, what each of the files you wrote for the project contains and does, and if you debated certain design choices, explaining why you made them. Ensure you allocate sufficient time and energy to writing a `README.md` that documents your project thoroughly. Be proud of it! A `README.md` in the neighborhood of 750 words is likely to be sufficient for describing your project and all aspects of its functionality. If unable to reach that threshold, that probably means your project is insufficiently complex.

Execute the `submit50` command below from within your `project` directory (or from whichever directory contains `README.md` file and your project's code, which must also be submitted), logging in with your GitHub username and password when prompted. For security, you'll see asterisks instead of the actual characters in your password.

```
submit50 cs50/problems/2024/x/project
```

▶ **Trouble Submitting?**

Step 3 of 3

Be sure to visit your gradebook at cs50.me/cs50x (https://cs50.me/cs50x) a few minutes after you submit. It's only by loading your Gradebook that the system can check to see whether you have completed the course, and that is also what triggers the (instant) generation of your free CS50 Certificate and the (within 30 days) generation of the Verified Certificate from edX, if you've completed all of the other assignments. Be sure to claim your free certificate (by following the link at the top of your gradebook) before 1 January 2025.

> Don't skip the above step! The course is not considered complete until you do the above and see the green banner saying you've completed the course. If you do not do the above prior to 1 January 2025, your status in the course will be subject to the **carryover rules** in the FAQ. The staff will not make any manual corrections in early 2025 based on this being skipped!

That's it! Your project should be graded within a few minutes. If you don't see any results in

your gradebook, best to resubmit (running the above `submit50` command) with only your README.md file this time. No need to resubmit your form.

This was CS50x!