

---

# Final Project

This is CS50. Harvard University. Fall 2015.

## Table of Contents

Overview .....	1
Academic Honesty .....	3
Reasonable .....	4
Not Reasonable .....	5
Assessment .....	6
Schedule .....	6
Pre-Proposal .....	6
Proposal .....	7
Status Report .....	7
CS50 Hackathon .....	7
Implementation .....	7
CS50 Fair .....	7
Pre-Proposal .....	7
Proposal .....	8
Status Report .....	8
CS50 Hackathon .....	8
Implementation .....	9
How to Submit .....	10
Step 1 of 2 .....	10
Step 2 of 2 .....	11
CS50 Fair .....	11

## Overview

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 infrastructure other than CS50 IDE, 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.

If taking some other course this semester that has a final project, you are welcome and encouraged to combine this course's project and that course's project into one, toward an end of applying lessons learned in CS50 to some other field, so long as the joint project satisfies this course's and that course's expectations. Before pursuing a joint project, though, you must disclose to both courses and receive approval from both courses (as from your TF in CS50).

Moreover, inasmuch as software development is rarely a one-person effort, you are allowed an opportunity to collaborate with one or two other students 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 any of the course's problem sets.

If at a loss for ideas, turn to the below (and your TF) for inspiration!

- [CS50 Seminars](#)<sup>1</sup>
- [CS50 Discuss](#)<sup>2</sup>

CS50 also hosts a number of APIs that you're welcome to integrate into your own project:

- [CS50 Courses API](#)<sup>3</sup>
- [CS50 Food API](#)<sup>4</sup>
- [CS50 Maps API](#)<sup>5</sup>

Know, too, that CS50 has:

---

<sup>1</sup> <https://cs50.net/seminars>

<sup>2</sup> <http://newapps.cs50.net/cs50/@/discuss/threads/136/all/new/1>

<sup>3</sup> <https://manual.cs50.net/api/courses/>

<sup>4</sup> <https://manual.cs50.net/api/food/>

<sup>5</sup> <https://manual.cs50.net/api/maps/>

- an [iOS Developer Program](https://developer.apple.com/programs/)<sup>6</sup>, which allows you to install homemade apps on your own iOS device,
- some [Arduino Uno](https://www.arduino.cc/en/Main/arduinoBoardUno)<sup>7</sup> boards, for which there's an [API](https://www.arduino.cc/en/Reference/HomePage)<sup>8</sup>,
- some [Google Cardboards](https://www.google.com/get/cardboard/)<sup>9</sup>, for which there's an [developers.google.com/cardboard\[API\]](https://developers.google.com/cardboard[API]),
- some [Leap Motion Controllers](https://www.leapmotion.com/)<sup>10</sup>, for which there's an [API](https://developer.leapmotion.com/)<sup>11</sup>,
- some [Microsoft Bands](http://www.microsoft.com/microsoft-band/en-us)<sup>12</sup>, for which there's an [SDK](https://developer.microsoftband.com/)<sup>13</sup>
- some [Myo Armbands](https://www.myo.com/)<sup>14</sup>, for which there's an [API](https://developer.thalmic.com/)<sup>15</sup>,
- some [Philips hue bulbs](http://www2.meethue.com/en-us/productdetail/philips-hue-white-and-color-ambiance-starter-kit-a19)<sup>16</sup>, for which there's an [API](http://www.developers.meethue.com/)<sup>17</sup>,
- some [Philips hue lightstrips](http://www2.meethue.com/en-us/productdetail/philips-hue-lightstrip)<sup>18</sup>, for which there's an [API](http://www.developers.meethue.com/)<sup>19</sup>, and
- some [Samsung Gear VRs](http://www.samsung.com/global/microsite/gearvr/)<sup>20</sup>, for which there's an [SDK](http://developer.samsung.com/gear/)<sup>21</sup>.

If you'd like to borrow something for a project, email [borrow@cs50.harvard.edu](mailto:borrow@cs50.harvard.edu)<sup>22</sup>, and we'll do our best to oblige, sharing hardware across classmates as needed. Let us know, too, if there's some other hardware or software you'd like to utilize for a project, and we'll do our best to find it.

## 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

---

<sup>6</sup> <https://developer.apple.com/programs/>

<sup>7</sup> <https://www.arduino.cc/en/Main/arduinoBoardUno>

<sup>8</sup> <https://www.arduino.cc/en/Reference/HomePage>

<sup>9</sup> <https://www.google.com/get/cardboard/>

<sup>10</sup> <https://www.leapmotion.com/>

<sup>11</sup> <https://developer.leapmotion.com/>

<sup>12</sup> <http://www.microsoft.com/microsoft-band/en-us>

<sup>13</sup> <https://developer.microsoftband.com/>

<sup>14</sup> <https://www.myo.com/>

<sup>15</sup> <https://developer.thalmic.com/>

<sup>16</sup> <http://www2.meethue.com/en-us/productdetail/philips-hue-white-and-color-ambiance-starter-kit-a19>

<sup>17</sup> <http://www.developers.meethue.com/>

<sup>18</sup> <http://www2.meethue.com/en-us/productdetail/philips-hue-lightstrip>

<sup>19</sup> <http://www.developers.meethue.com/>

<sup>20</sup> <http://www.samsung.com/global/microsite/gearvr/>

<sup>21</sup> <http://developer.samsung.com/gear/>

<sup>22</sup> <mailto:borrow@cs50.harvard.edu>

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 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.

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 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.

## Assessment

Your pre-proposal, proposal, and status report will be evaluated on the bases of, at least, clarity and thoroughness. Your implementation will be evaluated along four axes primarily:

### Scope

To what extent does your code implement the features required by our specification?

### Correctness

To what extent is your code consistent with our specifications and free of bugs?

### Design

To what extent is your code written well (i.e., clearly, efficiently, elegantly, and/or logically)?

### Style

To what extent is your code readable (i.e., commented and indented with variables aptly named)?

All students, whether or not taking the course for a letter grade, must ordinarily submit this final project to be eligible for a satisfactory grade unless granted an exception in writing by the course's heads.

## Schedule

### 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*

Extensions on the final project are not granted, except in cases of emergency. Technical difficulties are not considered emergencies. Submitting more than seven minutes late is equivalent to not submitting at all.

## Pre-Proposal

Intended to promote early thought, the pre-proposal is your opportunity to bounce one or more ideas off of your teaching fellow. Quite simply, by this pre-proposal's deadline, send an email to your teaching fellow, CCing [projects@cs50.harvard.edu](mailto:projects@cs50.harvard.edu)<sup>23</sup>, describing one or more ideas that you have for your final project. Short, casual emails are fine, but do explain the motivation behind each of your ideas (i.e., why it interests you). Treat this requirement as an opportunity for counsel. Certainly include any questions or concerns that you have in this email.

The subject line of your email should be **Pre-Proposal**.

---

<sup>23</sup> <mailto:projects@cs50.harvard.edu>

If, incidentally, you have an idea for a final project that you think someone should do (if not you), post it [CS50 Discuss](#)<sup>24</sup>! And if you'd like to solicit one or two collaborators, do post there as well.

## Proposal

The proposal is your opportunity to receive approval and counsel from your teaching fellow before you proceed to design. Submitting a proposal amounts to answering a few questions about your idea. Once you have a project in mind, submit your proposal at [cs50.harvard.edu/proposal](https://cs50.harvard.edu/proposal)<sup>25</sup>.

Your teaching fellow will either approve your proposal or require modifications on your part for subsequent approval. Your proposal, even if approved, is not binding; you may alter your plan at any point, provided you obtain the staff's approval for any modifications. Projects submitted without approval may not receive credit.

After submitting your proposal, a teaching fellow other than your own may be appointed your advisor and grader for the remainder of the final project, depending on your proposal's nature.

## Status Report

Not only is the status report intended to keep the staff apprised of your progress, it is an opportunity to keep yourself on track. Submitting a status report amounts to answering a few questions about your project. Your answers will also influence the setup of the CS50 Fair. Submit your status report at [cs50.harvard.edu/project/report](https://cs50.harvard.edu/project/report)<sup>26</sup>.

## CS50 Hackathon

The CS50 Hackathon is an epic all-nighter during which you can dive into your final project's implementation alongside classmates 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.

---

<sup>24</sup> <http://newapps.cs50.net/cs50/@/discuss/threads/136/all/new/1>

<sup>25</sup> <https://cs50.harvard.edu/proposal/>

<sup>26</sup> <https://cs50.harvard.edu/project/report/>



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.

Additional details on the CS50 Hackathon's logistics will be announced via email and the course's home page the week before the CS50 Hackathon.

## Implementation

Ultimately due are implementation and documentation of your final project. Your submission thereof must include all of the below.

1. Documentation for your project in the form of a file called `documentation.html`, `documentation.pdf`, `documentation.php`, or `documentation.txt`. This documentation is to be a user's manual for your project. Though the structure of your documentation is entirely up to you, it should be incredibly clear to the staff how and where, if applicable, to compile, configure, and use your project. Your documentation should be at least several paragraphs in length. It should not be necessary for us to contact you with questions regarding your project after its submission. Hold our hand with this documentation; be sure to answer in your documentation any questions that you think we might have while testing your work.
2. A "design document" for your project in the form of a file called `design.html`, `design.pdf`, `design.php`, or `design.txt` that discusses, technically, how you implemented your project and why you made the design decisions you did. Your design document should be at least several paragraphs in length. Whereas your documentation is meant to be a user's manual, consider your design document your opportunity to give the staff a technical tour of your project underneath its hood.
3. Any and all files required to run your software (even if intended for some infrastructure other than the CS50 Appliance), including source code as well as, if applicable, configuration files, Makefiles, sample inputs, and so forth. Needless to say, all source code should be thoroughly commented. If your project uses a MySQL database, be sure to export it to a file (e.g., `project.sql`), as with phpMyAdmin's **Export** tab, and include that file in the directory that you submit.
4. A short video (that's no more than 2 minutes in length) in which you present your project to the world, as with slides, screenshots, voiceover, and/or live action. Your video should somehow include your project's title, your name and year, your dorm/house and

concentration, and any other details that you'd like to convey to viewers. See <http://www.cs171.org/2015/screencast/> for CS171's tips on how to make a "screencast," though you're welcome to use an actual camera. Upload your video to YouTube as "public" or "unlisted" and take note of its URL.

## How to Submit

**If you have collaborated with one or two other students, each of you should submit via this same process.**

**If your project requires (for execution and testing) hardware or software other than that offered by the CS50 Appliance, be sure that the TF advising you is aware of and has approved your project's needs.**

### Step 1 of 2

- Create a ZIP (i.e., compressed) file containing your entire project, including all files prescribed above.
- Once done creating your ZIP file, open up a browser and visit [CS50 Submit<sup>27</sup>](#), logging in if prompted.
- Click **Submit** toward the window's top-left corner.
- Under **Implementation** on the screen that appears, click **Upload New Submission**.
- On the screen that appears, click **Add files....** A window entitled **Open Files** should appear.
- Navigate your way to your project's ZIP file. Once you find it, click it once to select it, then click **Open**.
- Click **Start upload** to upload your ZIP file to CS50's servers.
- On the screen that appears, you should see a window with **No File Selected**. If you move your mouse toward the window's lefthand side, you should see a list of the files you uploaded. Click each to confirm the contents of each. (No need to click any other buttons or icons.) If confident that you submitted the files you intended, consider your source code submitted! If you'd like to re-submit different (or modified) files, simply

---

<sup>27</sup> <https://cs50.net/submit>

return to [CS50 Submit](#)<sup>28</sup> and repeat these steps. You may re-submit as many times as you'd like; we'll grade your most recent submission, so long as it's before the deadline.

## Step 2 of 2

Head to <https://forms.cs50.net/2015/fall/project/implementation> where a form (your last!) awaits. Once you have submitted that form (as well as your source code), you are done!

## CS50 Fair

The CS50 Fair is an epic display of final projects, your opportunity to showcase your work not only to us but also to others on campus. You will be expected to bring to the CS50 Fair a laptop with which to demonstrate your project. Plan to tell attendees what you have done and why you have done it. And perhaps have in mind a few anecdotes about lessons you learned, roadblocks you hit, or the like.

The CS50 Fair will take place in the atrium of Northwest Science Labs at 52 Oxford Street.

Additional details on the CS50 Fair's logistics will be announced via email and the course's home page the week before the CS50 Fair.

---

<sup>28</sup> <https://cs50.net/submit>