# Writing Problem 0-0: Around the House

This is CS50. Harvard University. Fall 2014.

## Table of Contents

## Objectives

- Think critically about computers and computing.

- Learn about how to write with an eye towards a technical audience.

## 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 problems is not permitted (unless explicitly stated otherwise) 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 or writing to others, but you may not view theirs, so long as you and they respect this policy's other constraints. Collaboration on quizzes and tests is not permitted at all. Collaboration on the 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 your instructor. If a violation of this policy is suspected and confirmed, your instructor reserves the right to impose local sanctions on top of any disciplinary outcome that may include an unsatisfactory or failing grade for work submitted or for the course itself.

## Reasonable

- Communicating with classmates about 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, such 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 years' quizzes, tests, 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 solutions to problems online so that others might help you identify and fix a bug or other issue.

- 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 problems or your own final project.

- Whiteboarding solutions to problems 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 before (re-)submitting your own.

- Decompiling, deobfuscating, or disassembling the staff's solutions to problems.

- Failing to cite (as with comments) the origins of code, writing, 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 when it is he or she, and not you, who is struggling to solve it.

- Looking at another individual's work during a quiz or test.

- Paying or offering to pay an individual for work that you may submit as (part of) your own.

- Providing or making available solutions to problems 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 problems online or elsewhere.

- Splitting a problem's workload with another individual and combining your work (unless explicitly authorized by the problem itself).

- 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.

- Using resources during a quiz beyond those explicitly allowed in the quiz's instructions.

- Viewing another's solution to a problem and basing your own solution on it.

## Assessment

Your work on this writing problem will be evaluated along three axes primarily.

**Scope**

　To what extent does your submission align with the requirements of the specification?

**Correctness**

    To what extent is your submission correct and free of factual errors?

**Style**

    To what extent is your submission readable (i.e., thoughtfully organized, coherent, words properly spelled)?

To obtain a passing grade in this course, all students must ordinarily submit all assigned problems unless granted an exception in writing by the instructor.

# OK Computer

If not familiar with what computers are, take a quick look at a more modern spin on them here:

https://www.youtube.com/watch?v=WAxH0YHdTuA

Although they've also been known to be a bit smaller than that:

https://www.youtube.com/watch?v=LHdVkPrdRYg

But perhaps there are some that do not even use electricity?

https://www.youtube.com/watch?v=GcDshWmhF4A

Or perhaps may not even have moving parts?

https://www.youtube.com/watch?v=tI0GqYJha1Q

If you had some preconceived notions about what a computer is, odds are they were challenged somewhat by watching the videos above (particularly the last two!), and that's okay! In fact, you may find that some folks quickly agree that everything shown above is a computer, and some won't. Try speaking with family and friends about it and see if it doesn't spark an interesting discussion.

We'll wait here while you do that.

*(whistles, twiddles thumbs)*

O hai! You're back.

Pull up the definition of the word "computer" a few places online (or in a handy, printed dictionary if you happen to still have one!), and while it's likely you'll find some common threads, it's also quite likely that no two definitions are the same.

For purposes of this course, we define a computer as *a device that accepts input and processes it in some way to produce an output automatically*. Based on that definition, you might see how all four of the devices shown above may be considered computers. You also still may be scratching your head and thinking "Well, wait a minute…". If so, good! You're already beginning the critical-thinking process.

# Writing? I thought I was here to code!

Rest assured, there'll be plenty of time and plenty of opportunity to dive into programming this school year. In fact, the vast majority of the problems you're assigned in this course will require you to program in one or more of the programming languages we'll learn about, such as Scratch, C, PHP, or JavaScript. But occasionally, and a bit more frequently at the very beginning of the course, you'll also be asked to complete some "writing problems" like this. Why? There are two important reasons.

First and foremost, writing is an essential part of your assessment by the College Board for Advanced Placement credit in the course. The assessment consists of three parts:

- The end-of-year multiple-choice examination;

- A through-course assessment called "Create," in which you will independently and with a partner develop programs that solve real-world problems; and

- Another through-course assessment called "Explore," in which you will critically research and investigate an innovation in computing and the impact it has had on the global community.

All the programming problems you'll encounter in this course will adequately prepare you to tackle "Create," and the writing problems are similarly designed to prepare you for "Explore."

Second however, and perhaps more importantly, is that this course aims to educate you more broadly as a *computer scientist*, and not just specifically as a *computer programmer*. Being able to write code is just one tool at your disposal. Among the many characteristics of a computer scientist is his or her ability to communicate effectively with others, both with and without technological backgrounds alike, about computers, emerging technologies, and more. By researching these topics and reading about technology on tech news sites like TechCrunch, Wired, Engadget, and others, you'll not only become more conversant in

the jargon that computer scientists use to describe technology, computers, and computing, but you'll also improve in your ability to further relay what you've learned to others.

In some contexts you'll be relaying your newfound knowledge to those with absolutely no background in the subject matter, and to address those individuals you'll need the ability to describe things clearly and, importantly, correctly. In others, you'll be addressing your peers or more technically-oriented audiences, and instead of having to *explain* a new technology, you will have to *persuade* that audience about something. Being able to analyze a technology, compare it to others, and point out its relative flaws is an important rhetorical tool to do just that. Organizing your thoughts and communicating them on paper is one of the best ways (particularly if you otherwise might experience stage fright!) to practice this skill.

# OK… Computer?

Have a look around your home. See any computers? Even if you don't have laptops or desktops where you live, odds are you have many more computers in your home than you think. Perhaps you have a smartphone? Maybe you have a flatscreen television or a video game system?

In this problem, we want you to think even more outside the box than that. Recall that we defined a computer as *a device that accepts input, and processes it in some way to produce a result automatically*. Surely there exists some device in your home that adheres to that definition but isn't something that before now you would have readily called a computer. If you can't think of any such device in your home, feel free to venture beyond those four walls and pick any device with which you may be familiar.

In no more than 400 words[1], describe this device in detail. You may wish to consider questions such as:

- What does the device look like?

- What kind of data does it accept?

- How does it process that data?

---

[1] Seriously! In the real world, projects often have specifications just like this one, and it's frequently quite important to adhere to those specifications exactly so that you are in compliance with project scope. So keep it to 400 words, tops!

- What is the result of that processing?

Conduct a little bit of research (formal or informal), and if you suspect others may be skeptical of your assertion that the device you chose is a computer, write persuasively so as to do your best to convince such individuals that you aren't crazy and that you know what you're talking about.

This was Writing Problem 0-0, your first problem in CS50 AP!