Problem Set 5: Forensics

due ~~Thu 10/24~~ Fri 10/25 at noon

No need to submit any work early (i.e., by Wed) this week in order to get an extra day. Everyone gets an extra day this week; everyone's deadline is Fri 10/25. No more coupon codes!

Questions? Head to cs50.net/discuss (https://www.cs50.net/discuss) or join classmates at office hours (https://www.cs50.net/ohs)!

Objectives

- Acquaint you with file I/O.
- Get you more comfortable with data structures, hexadecimal, and pointers.
- Introduce you to computer scientists across campus.
- Help Mr. Boddy.

Recommended Reading*

- Chapters 18, 24, 25, 27, and 28 of *Absolute Beginner's Guide to C*
- Chapters 9, 11, 14, and 16 of *Programming in C*
- http://www.cprogramming.com/tutorial/cfileio.html (http://www.cprogramming.com/tutorial/cfileio.html)
- http://en.wikipedia.org/wiki/BMP_file_format (http://en.wikipedia.org/wiki/BMP_file_format)
- http://en.wikipedia.org/wiki/Hexadecimal (http://en.wikipedia.org/wiki/Hexadecimal)
- http://en.wikipedia.org/wiki/Jpg (http://en.wikipedia.org/wiki/Jpg)

* The Wikipedia articles are a bit dense; feel free to skim or skip!

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.

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 on CS50 Discuss or elsewhere 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 in CS50 Vault 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.
- 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 your 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.
- Redeeming or attempting to redeem someone else's code for a late day.
- 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.

Scores

Your work on this problem set 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 taking the course SAT/UNS or for a letter grade, must ordinarily submit this and all other problem sets to be eligible for a satisfactory grade unless granted an exception in writing by the course's heads.

Shorts

- Head to https://www.cs50.net/shorts/7 (https://www.cs50.net/shorts/7) and watch the shorts on File I/O, Structs, and Valgrind. Just keep in mind that Jason's short on File I/O happens to focus on ASCII (i.e., text) files as opposed to binary files (like images). More on those later!
- You may also want to re-watch the short on GDB at https://www.cs50.net/shorts/4! (https://www.cs50.net/shorts/4!)

Getting Started

- Welcome back!
- As always, first open a terminal window and execute

```
update50
```

to make sure your appliance is up-to-date.

- Like Problem Set 4, this problem set comes with some distribution code that you'll need to download before getting started. Go ahead and execute

  ```
  cd ~/Dropbox
  ```

  in order to navigate to your `~/Dropbox` directory. Then execute

  ```
  wget http://cdn.cs50.net/2013/fall/psets/5/pset5/pset5.zip
  ```

  in order to download a ZIP (i.e., compressed version) of this problem set's distro. If you then execute

  ```
  ls
  ```

  you should see that you now have a file called `pset5.zip` in your `~/Dropbox` directory. Unzip it by executing the below.

  ```
  unzip pset5.zip
  ```

  If you again execute

  ```
  ls
  ```

  you should see that you now also have a `pset5` directory. You're now welcome to delete the ZIP file with the below.

  ```
  rm -f pset5.zip
  ```

  Now dive into that `pset5` directory by executing the below.

  ```
  cd pset5
  ```

  Now execute

  ```
  ls
  ```

  and you should see that the directory contains the below.

  ```
  bmp/  jpg/  questions.txt
  ```

How fun! Two subdirectories and a file. Who knows what could be inside! Let's get started.

- If you ever saw Windows XP's default wallpaper (think rolling hills and blue skies), then you've seen a BMP. If you've ever looked at a webpage, you've probably seen a GIF. If you've ever looked at a digital photo, you've probably seen a JPEG. If you've ever taken a screenshot on a Mac, you've probably seen a PNG. Read up a bit on the BMP, GIF, JPEG, and PNG file formats. Then, open up `questions.txt` in `~/Dropbox/pset5`, as with `gedit`, and tell us the below.

  0. How many different colors does each format support?
  1. Which of these formats supports animation?
  2. What's the difference between lossy and lossless compression?
  3. Which of these formats is lossy-compressed?

- Curl up with the article from MIT at http://cdn.cs50.net/2013/fall/psets/5/garfinkel.pdf (http://cdn.cs50.net/2013/fall/psets/5/garfinkel.pdf).

  Though somewhat technical, you should find the article's language quite accessible. Once you've read the article, answer each of the following questions in a sentence or more in `~/Dropbox/pset5/questions.txt`.

  4. What happens, technically speaking, when a file is deleted on a FAT file system?
  5. What can someone like you do to ensure (with high probability) that files you delete cannot be recovered?

Whodunit and more

- Welcome to Tudor Mansion. Your host, Mr. John Boddy, has met an untimely end—he's the victim of foul play. To win this game, you must determine `whodunit`.

  Unfortunately for you (though even more unfortunately for Mr. Boddy), the only evidence you have is a 24-bit BMP file called `clue.bmp`, pictured below, that Mr. Boddy whipped up on his computer in his final moments. Hidden among this file's red "noise" is a drawing of `whodunit`.