

CSC 372, Spring 2014
Assignment "V" (for video!)
Due: Wednesday, May 7 at 23:00

In a nutshell

The essence of this assignment is simple:

- (1) Find in some programming language an interesting element or aspect that you know little about.
- (2) Experiment with that interesting thing and see what you can learn about it.
- (3) Make a 10-minute video of yourself talking about that interesting thing.

This assignment is worth 40 points—4% of your final grade.

The final Prolog assignment will also be due on May 7, so get an early start on this assignment and spread out the work over the next five weeks.

Finding a language

A key requirement is that you need to pick a topic that you know little about. In a language you don't know, any aspect of the language would qualify although many might not be very interesting, like straightforward analogs of Java control structures and data types. I wouldn't want you talking about iterators in Ruby, but Ruby threads would certainly be fair game. If you feel like taking on a beast, you might try monads in Haskell. Should you want to do something in Prolog, ok the topic with me first, to be sure it won't be something we'll be covering.

If you want to work with a language that's new to you, there are thousands to choose from! You'll probably want to choose a language that has an implementation available for your machine or is installed on lectura so that you can experiment with it. A language with an interactive mode (a REPL) makes experimentation easier, of course.

Slide 20 in the intro set lists a number of prominent languages. Lots of choices, maybe too many, can be found at http://en.wikipedia.org/wiki/List_of_programming_languages A good middle ground is http://dir.yahoo.com/computers_and_internet/programming_and_development/languages. Some older languages have quite a few features that are fairly different from what we commonly see today and might present some low hanging fruit. COBOL, Forth, MUMPS, and (my favorite old language) SNOBOL4 come to mind.

Finding an interesting thing

In elementary school you might have written a report on "The Civil War". By the time you got to college you probably realized that "The Civil War" was a hopelessly big topic to cover in a single report. But maybe "The Battle of Picacho Peak" would have been about the right size.

Similarly here, **the task is not to cover the full language but just one single aspect or element**. Had we not covered these topics in class, each of the following would make a fine ten-minute topic:

- Type inferencing in Haskell
- Curried functions and partial evaluation in Haskell
- Computations with map, filter, and fold
- Ruby strings, arrays, or hashes
- Ruby iterators contrasted with conventional iteration

Here are some ten-minute topic ideas:

- More with infinite lists in Haskell
- Functors in Haskell
- Ruby's `ObjectSpace`
- Contrast regular expressions in Java vs. Ruby
- Some Ruby regular expression operators that we didn't talk about
- `method_missing` in Ruby
- List comprehensions in Python
- "Globals" in MUMPS
- Simple pattern matching in SNOBOL4
- `PICTURE` in COBOL
- Upcoming lambda support in Java
- Prototypes in JavaScript
- Ruby's `Matrix` class
- Control structures in `bash`
- Arrays in `bash` (ick!)
- Datatypes in BCPL
- Python dictionaries
- PHP arrays
- Stack-based programming in PostScript or Forth
- How ML describes types vs. Haskell's notation
- Using C's `sizeof` to explore the language
- Fun with the C preprocessor

One way to look for topic ideas is to browse tables of contents for books on Safari. For high-bandwidth browsing, go to the library.

The goal here is technical entertainment. Your challenge is to find some interesting aspect or element with which you think you can create material that will hold somebody's attention for ten minutes.

You can make any assumptions you want about your audience. For example, you might assume that your audience is your 372 classmates, implying that they know Java, Ruby, Haskell, and, by the end, Prolog. If you want to talk about the C preprocessor, you should assume the audience knows some C.

It's fine to use examples you discover on the web and in books; just take a moment to verbally cite the source, like the author's name, the name of the book, or the website. It's fine to watch videos to help find a topic or learn more about your topic.

You don't need to get your topic approved in advance but I'll be happy to offer advice on your ideas, or help you find an idea.

The video

If your language has an interactive mode like Haskell, Ruby, and Prolog, your video might just be a screen recording of you demonstrating a feature and narrating what you're doing. I like Jing for screen recordings that but there's a five-minute limit (do it in two parts).

If you're inclined to write slides with PowerPoint, Keynote, or whatever, and record a presentation, perhaps even projected on a screen in a classroom, that's fine.

You can work with pencil and paper, writing out examples by hand, as one might when working on an

Elmo, and shoot video with a cell phone or a camcorder.

Any mix of the above modes or additional modes is fine.

Try a test video well before the deadline. For example, if you're planning to use your cell phone to shoot video, be sure the resolution is adequate to make text legible and that your phone has enough space for whatever length segments you plan to shoot.

Between screen recordings, cell phones, and digital cameras I imagine most everybody will already have the equipment they need but if not, <http://www.library.arizona.edu/services/equipment-lending> and <http://www.uits.arizona.edu/departments/oscr/locations/gtg> have loaner cameras and tripods.

I don't expect you to spend time editing video to make it perfect, or anywhere near perfect. You're certainly free to edit if you want, but I'm fine with a single "take".

It's fine to run a little past ten minutes, but keep it under eleven minutes. If you're significantly under ten minutes there will be a proportional deduction in your grade.

Turning in your work

You can park your video on the web or deliver a DVD to me, but either way submit a plain text file named `video.txt` to the aV Dropbox on D2L. The file should be structured as follows:

First line: A title for your video

Second line: A URL for the video (or just say "DVD")

Third line: "Post: yes" or "Post: no", if a URL.

Following lines: Any sort of comments or observations you'd like to make, if any.

I'll collect the URLs for those which are "Post: yes" and post that list on Piazza. I'll use the title specified. If you want your name or a pseudonym shown, include it in the title, maybe with "... by John Smith" or "The Amazing Haskellon Explores Monads", for example.

Putting your video on the web and specifying "Post: yes" earns you a five-point bonus but you can remain anonymous, except to those who might recognize your voice, for example.

Late submissions accepted

Unlike other assignments, late submissions will be accepted on this assignment, with a penalty of two points per 24 hours or any fraction thereof, with a maximum of five days (5*24 hours) late.

Expectations

I'm picturing that a typical student will spend 6-8 hours on this assignment.

I don't expect you to achieve full understanding of your topic; you just need to know enough to fill ten minutes. It's fine to point out some things that you were unable to figure out.

If you have a disability that makes this assignment difficult for you, please let me know; we'll work out an accommodation.