CSSE 304 Assignment 0

Ok the main goal of this assignment is to get your development environment setup. This term we’re using a new version of scheme than we have in years past called “Racket” – this a flavor of scheme that is mostly compatible with standard scheme but has a couple of more modern tweaks. The reason we’re using is that it has much more robust debugging capability than classic scheme.

You should be able to download it here: <https://download.racket-lang.org/>

On my system windows complained about an unverified executable – it should be safe to “Run Anyway” to start the installer. Once installed there will be at least 2 apps one called Racket and called DrRacket. DrRacket is the one you’ll want to use – it is the IDE for Racket.

I also recommend you clone a local copy of [the course repo](https://github.com/RHIT-CSSE/csse304) (click the green “Code” button on that page and select your preferred download method). You won’t submit your assignments via git but it’s nice to get a local version with all the content you’ll need.

Watch [this video where I show some basics about running code in Racket](https://rose-hulman.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=ef292e55-7183-4fb9-a51d-ade90150fece).

Take the already written code in A0.rkt, run in your scheme interpter to verify it works, and them **submit it Gradescope for grading** (Gradescope link on the course Moodle).

If this works well your environment is setup and you’re done! You might consider moving on to the first real assignment A1 which is due tomorrow.

**OPTIONAL ADDENDIUM**

If you’re craving something a little more wild on the first day based on CSSE304’s reputation, perhaps you’re unfamiliar with the concept of a “quine”? Here’s one in scheme:

((lambda (f) (display (list f (list 'quote f)))) '(lambda (f) (display (list f (list 'quote f)))))

To see it work, run that in your Dr. Scheme environment. Then take the output and run that. 😊 What’s going on – this is a program that outputs its own code – i.e. it’s a quine. It relies on only 1 tiny trick that (display (list 'quote (list 1 2))) doesn’t print the way you would expect. Beyond that, you should have most of what you need to understand how it works after 1 or 2 days lecture. See if you can figure out how it works, and maybe modify it and make your own variant. If you find one you like, feel free to email me it. No extra credit for this, just the joy of playing with a weird computer science thing.

If it seems opaque to you now, come back in a couple weeks once you get familiar with the Scheme language. It’ll get better.

For more on quines, <http://conway.rutgers.edu/~ccshan/wiki/blog/posts/Quines/>

This page has some quines in Java which you might illuminate the concept a little more clearly [https://en.wikipedia.org/wiki/Quine\_%28computing%29](https://en.wikipedia.org/wiki/Quine_%28computing%29%20)