

# CSSE 304 Assignment 0 Updated for Fall, 2019-20

**No late days may be used or earned for this assignment**

## Summary of Deliverables (more details below)

- Submit the 0.ss file to the PLC server. Due **first day of class** at 11:59 PM. (see step 5 on page 2 of this document).
- Go to the CSSE 304 Piazza class by **first day of class** at 11:59 PM. Set up your email preferences.  
<https://piazza.com/rose-hulman/fall2019/csse304>.
  - I suggest setting your email notifications to "Real time" for new questions and updates, and perhaps also check "Automatically follow ...". To get to these settings, click on the gear at the far right of the Piazza bar at the top of the page.
- Bring your completed hand-in sheet (Word or PDF) to the **second class meeting**. You may
  - complete it electronically and print it, or
  - print it and complete it by hand.

Word and PDF versions of the hand-in sheet are linked from the schedule page.

Abbreviations for the textbooks	EOPL	- Essentials of Programming Languages, 3rd Edition
	TSPL	- The Scheme Programming Language, 4 <sup>th</sup> Edition (available free on <a href="http://scheme.com">scheme.com</a> )
	EoPL-1	- Essentials of Programming Languages, 1 <sup>st</sup> Edition (the small 4-up excerpt, handed out in class; it is also on <a href="#">Moodle</a> )

In case you are unable to follow some of the links from this document, most of the files are in the [http://www.rose-hulman.edu/class/csse/csse304/201920/Homework/Assignment\\_0](http://www.rose-hulman.edu/class/csse/csse304/201920/Homework/Assignment_0) folder.

All of the textbooks have numerous exercises. You should read and give at least a little bit of thought to many of the exercises in the chapters that we cover, and actually work out as many as time allows. Some of the exercises contain information that is crucial to understanding the later material in the text. Of course I won't require you to hand in all of the exercises in the books, but you should think about them (in addition to helping you learn the material, some of them make excellent questions for the exams!).

Student assistants will be available in Percopo or the CSSE lab (F-217) at specified times (afternoons and/or evenings). The assistants can help you with installation, programming environment, language concepts, Scheme details, and specific issues from the assignments. A link to the list of available times will be posted on the schedule page (Session 2). I hope that these lab hours can begin the evening before class Day 1. I hope to have the entire first week's TA schedule posted before the Day 2 class meeting.

If you have trouble installing and using *Chez Scheme*, or if you don't understand something that you read about Scheme or encounter when you experiment with Scheme, the assistant lab hours are one possible source of help. You can also get help there with assignments or the course in general. I hope to have one or more TAs in F217 the evening before classes start – 7:00-10:00 PM.

## Overview: Main activities for this assignment (in order of importance; you may actually do them in any order)

1. Experiment with Scheme.
2. Read about Scheme in TSPL or EOPL-1.
3. Bookmark the schedule page <http://www.rose-hulman.edu/class/csse/csse304> in your web browser, so you can go there with a single click. You will go there frequently this term.
4. Try out the grading program that allows you to check your code for correctness before it is due.
5. Install Scheme (and possibly Emacs) on your laptop.

As you do these activities, **complete the Assignment 0 Hand-in Sheet**, linked from the schedule page (Session 1) in Word and PDF forms. I will also hand out a printed copy in the Day 1 class. You can either print that sheet and fill it in by hand, or you can fill in the Word version and print it, whichever is easier for you.

## Detailed list of things to do:

1. Find and bookmark the course schedule page (the link is in the main activities section above, and also on Moodle), which is the page you'll use most often in this course. <http://www.rose-hulman.edu/class/csse/csse304> It contains links to most of the other course web pages. Browse around to become familiar with what is there. If you have not read the **Syllabus** (linked from the schedule page), do so now.
2. Install *Chez Scheme* on your computer. To get Petite *Chez Scheme* 8.4, go to <http://scheme.com/download/>. For version 9.5, go to <https://cisco.github.io/ChezScheme/> and scroll down to **Get Chez Scheme**. My suggestion is that you get both. The older

version, 8.4, is consistent with TSPL (4<sup>th</sup> edition) and the PLC grading server. If you get 8.4, I suggest the 32-bit non-threaded version. Read the instructions and then click the link to get the install file, which you should then run. Scroll through the same page to find versions for other OS's. For **Mac OSX installation**, see [this discussion](#).

3. Perhaps, look for a better editor than SWL or Notepad. I like Emacs a lot. There was a time when I strongly encouraged students to get it and learn it for this course. Now there are plenty of other editors out there that work well with Scheme. If you prefer a “keep your hands on the keyboard” editor that interacts well with Scheme, Emacs may be right for you. But it does take several hours of practice to get to the point where you can use it efficiently.
  4. To install the Gnu Emacs editor, see [Emacs Installation and Setup Instructions](#). Gnu Emacs is available from <http://ftp.gnu.org/pub/gnu/emacs>. For the **MS Windows version**, click the *windows* link near the top of the page, scroll way down, and choose **emacs-24.3-bin-i386.zip**. (or a later version if you prefer). Unzip the file somewhere, then go to the *bin* folder and run *addpm.exe*. There are also **OSX** and **Linux** versions. The [Emacs and Scheme](#) document describes the use of Emacs with Scheme (it needs some updating, but the basics are correct).
  5. [Here](#) is some info about running Petite *Chez* Scheme in the [Sublime Text](#) editor.
  6. At the end of this document (In the Piazza Q&A session) is some info about running Scheme in VSCode.
  7. **Experiment with Scheme.** Open a Scheme interpreter and enter some Scheme expressions, like those in <http://www.rose-hulman.edu/class/csse/csse304/202010/Live-in-class/> (file may not be there until after the day 1 class) and some from the textbooks. For each example, for each expression try to predict Scheme's response before you press Enter. The easiest way to access Scheme for beginners is to use SWL, which comes with Petite *Chez* Scheme 8.4. Just open SWL and start entering Scheme code.
  8. [TSPL](#) (*The Scheme Programming Language*, 4<sup>th</sup> Edition): Skim Chapter 1 and then carefully read first three sections of Chapter 2 of TSPL (<http://www.scheme.com/tspl4>). You may want to look at the Assignment 0 hand-in sheet before you read from the book so you'll know some things to be looking for. There are a few questions whose answers can be found in the reading. You should not expect to understand everything you read in Chapter 1 the first time you read it. But it will give you an idea of some of the programming language concepts we will be studying, and how they work out in Scheme.
- It is a good idea to actually type in and execute some of the Scheme expressions that you find in Chapter 2. Try variations of some of the Scheme expressions from the textbook. Predict what will happen, then evaluate the expression. If your prediction was wrong, that is an area that you will need to work to understand, and perhaps to discuss with me or with one or more classmates.
- You should also consider Exercises 2.2.1-2.2.8, 2.3.1, 2.4.1-2.4.3, and 2.5.1. You do not have to write them up or turn them in at this point. You can check your answers to most of them by entering them in Scheme and looking at the output.
- Try out the automatic grading server. This program is student-written (a 2009 junior project) and maintained. Download the `0.ss` file from the [Assignment\\_0](#) folder.
  - In your web browser, go to <https://plc.csse.rose-hulman.edu/>.
  - Bookmark that page; you will need to go there dozens of times this term to submit your code.
  - Login with your normal Rose-Hulman network password.
  - If the **A0** assignment tab is not already highlighted at the top, click it. Then click **Browse**, and browse to the `0.ss` file that you downloaded.
  - Click **Upload**. (If you get an error message, try clicking your browser's **Back** button to see the results. Sometimes then pressing **Regrade** fixes it. You should get a perfect score.
9. **Optional:** Get ahead by reading the first part of the [EOPL-1](#) excerpt (handed out in class, and also available on Moodle). Again, interactively try out some of the Scheme code there. If you don't get to read the first few pages before Day 2, do so before doing Assignment 1.
  10. **Optional:** Read the [Assignment 0 FAQ](#). It contains answers to some questions asked by students in the past.

**Coming:** Assignment 1 (**due Day 2 at 11:59 PM**) will include a few non-recursive procedures that you are to write, test, and submit electronically to the PLC server.

**Optional,** if you want to try Gnu Emacs (within the next few days):

Install the Gnu Emacs editor on your laptop computer (If you are running Linux or OSX, Emacs should come pre-installed with the OS). Instruction details are [here](#), my **emacs.el** file is [here](#).

**Optional:** Try the Emacs editor.

See the [Emacs and Scheme](#) document (a few things are obsolete but I am working on it) for details. Spend a few minutes on the tutorial; come back to it when you have time. Many of the commands may be found in the Emacs menus; you do not have to learn the special key sequences for them right away. You are not required to learn Emacs for this course, but I recommend that sometime soon you invest a couple of hours to see what it can do for you. I think that investing time now will give you a big payoff as the course progresses, and it will be a valuable tool throughout your programming career. If you do decide to learn Emacs, you should eventually learn the key sequences for common commands; they will save you time in the long run.

### **Reminder of deliverables:**

Complete the **Assignment 0 Hand-in Sheet** and turn in hard copy at the beginning of the Day 2 class.

Submit **0.ss** on the PLC server

Go to **piazza.com** (<https://piazza.com/rose-hulman/fall2019/csse304>) and configure your email settings for the CSSE 304 class.

## **Piazza questions and answers from previous terms**

### **zero points on 0.ss Submission**

I seem to be having some trouble with Assignment 0. I followed the directions for submitting the test assignment, but when I uploaded it I got zero points. I tried multiple times, but I haven't been able to get it to give me any points. The only odd thing I can see is that when I download it, it says 0.ss and when I upload it, it says 0.ss.txt could that be the problem? and how would I fix that?

You need to make sure that the file name really is 0.ss, not 0.ss.txt. Windows does not help with this, but you can fix it. By default, Windows does not show the file extension (the .txt in this case) in File Explorer and some other places. **To fix it:** open File Explorer (the program in the task bar that has the file folder icon). Click the View tab. All the way on the right side of the ribbon, click Options. The Folder Options dialog should pop up. Click the View tab. Uncheck "Hide extensions for known file types". Not relevant to this issue, but I also suggest that you check "Display full path in the title bar". Then click OK. The problem should be solved.

### **Installation problem: could not find SWL**

Hey all,

I just installed Petite Chez Scheme 9.5 using windows installer from <https://cisco.github.io/ChezScheme/>

After installation, I could not find the SWL editor that comes along with the package.

Could anyone please tell me where to find it?

Also, I noticed from the scheme website that SWL is no longer being maintained (<https://www.scheme.com/swl.html>). Does that mean that only version 8.5 or before contains SWL in their installers?

### **the instructors' answer,**

*where instructors collectively construct a single answer*

I expect that your assumption is correct. If you want to use SWL, you probably have to use it with version 8.4.

I hope you'll find a better editor that you like, but some students use SWL throughout the course.

And using 8.4 may be better than 9.5 for you, because that is the version the grading server uses.

*for lingering questions and comments*

#### Actions



Resolved



Unresolved



**A student**

2 months ago

If you are looking for another editor VS Code has a really nice extension to color parenthesis, it has already helped me catch multiple errors in my code.



**Claude Anderson**

2 months ago

What is the name of the extension?



**Another student** 2 months ago

They have a vscode-scheme by Allen Huang. Haven't tried it myself, but might be useful if the default features of VS code aren't enough.

<https://marketplace.visualstudio.com/items?itemName=sjhuangx.vscode-scheme>

Just wanted to let you know if anyone might benefit from it.

#### Actions



**A student**

2 months ago

vscode-chex is the one I've been using

Yet another student:

Thank you Dr. Anderson!

As an alternative, I installed Sublime Text 3 with a Scheme syntax highlighter according to the installation guide you gave: <https://www.rose-hulman.edu/class/cs/csse304/202010/Resources/SublimeText3.pdf>