

Computer Science 1 — CSCI 1100

Lab 1 — Variables, Assignments, Expressions and Submittity

Fall Semester 2019

Lab Overview

This lab has two goals: (1) to learn how to use *Submittity* the submission server we will use for submitting homeworks and checking grades, and (2) to practice writing short programs that involve variables, assignments, expressions and simple string operations. This will be accomplished by three small checkpoints. Throughout, we will also practice finding and fixing syntax and semantic errors, begin our study of good program structure and also learn what hard-coding is and how to avoid it.

Checkpoint 1: Introduction to Submittity

For each set of lecture exercises and for each homework assignment you will log into Submittity, upload your code programs, run them, and view the results. For checkpoint one we will concentrate on submitting a single Python file in order to get a correctly running program with no syntactic or semantic errors.

Go to the Class Materials section on Submittity and download the program `lab1_volume.py`. (Store this in a folder you create within your Dropbox folder structure for Lab 1.) Bring up the Spyder IDE and Open this file (click File and then Open and navigate to the desired file) so you can look at the program closely. Rather than making you write your own program for this first part, we will start with this “flawed” version just to demonstrate some important aspects of *Submittity*. Note that this file is intended to calculate the volume of a cone. If you are fuzzy about the correct formula, it is $\frac{1}{3} * \text{area_of_base} * \text{height}$. I am going to assume you can calculate the area of the circular base!

This simple Python program contains **both** syntax and semantic errors. Don’t fix these errors yet, even if you know what they are! If you don’t remember the difference between a syntax error and a semantic error, look back at the Lecture 2 notes.

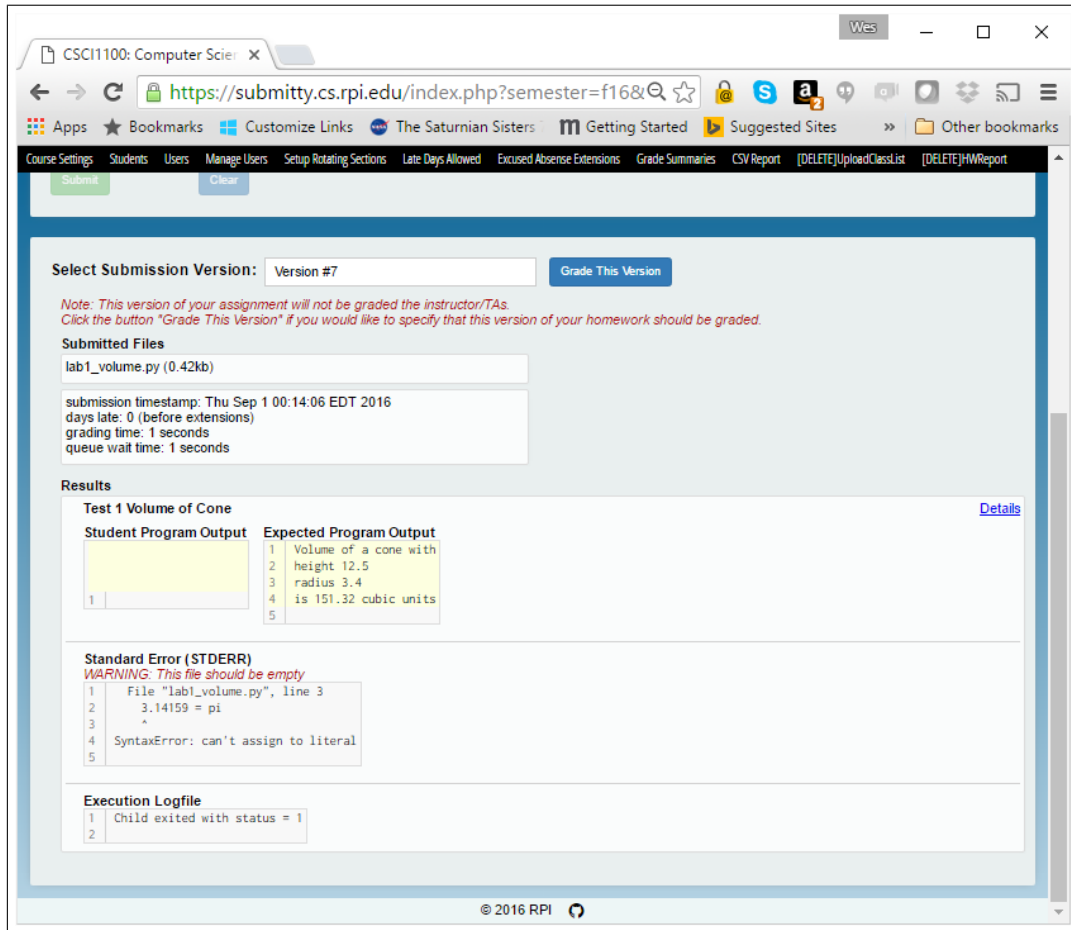
First, we want to familiarize you with the various errors messages you will get when you run your own program in the Spyder IDE and when you upload a program to Submittity that has errors. Please follow these steps:

- Look at `lab1_volume.py` in the IDE and, without making any changes, attempt to run it (click the green triangle/arrow). It will complain of syntax error(s). Do **NOT** fix these errors yet. Instead, upload this file as is to Submittity to see what happens.
- To do this, cut-and-paste the following link to the submission page into a browser (or click below)

<https://submittity.cs.rpi.edu/f19/csci1100>

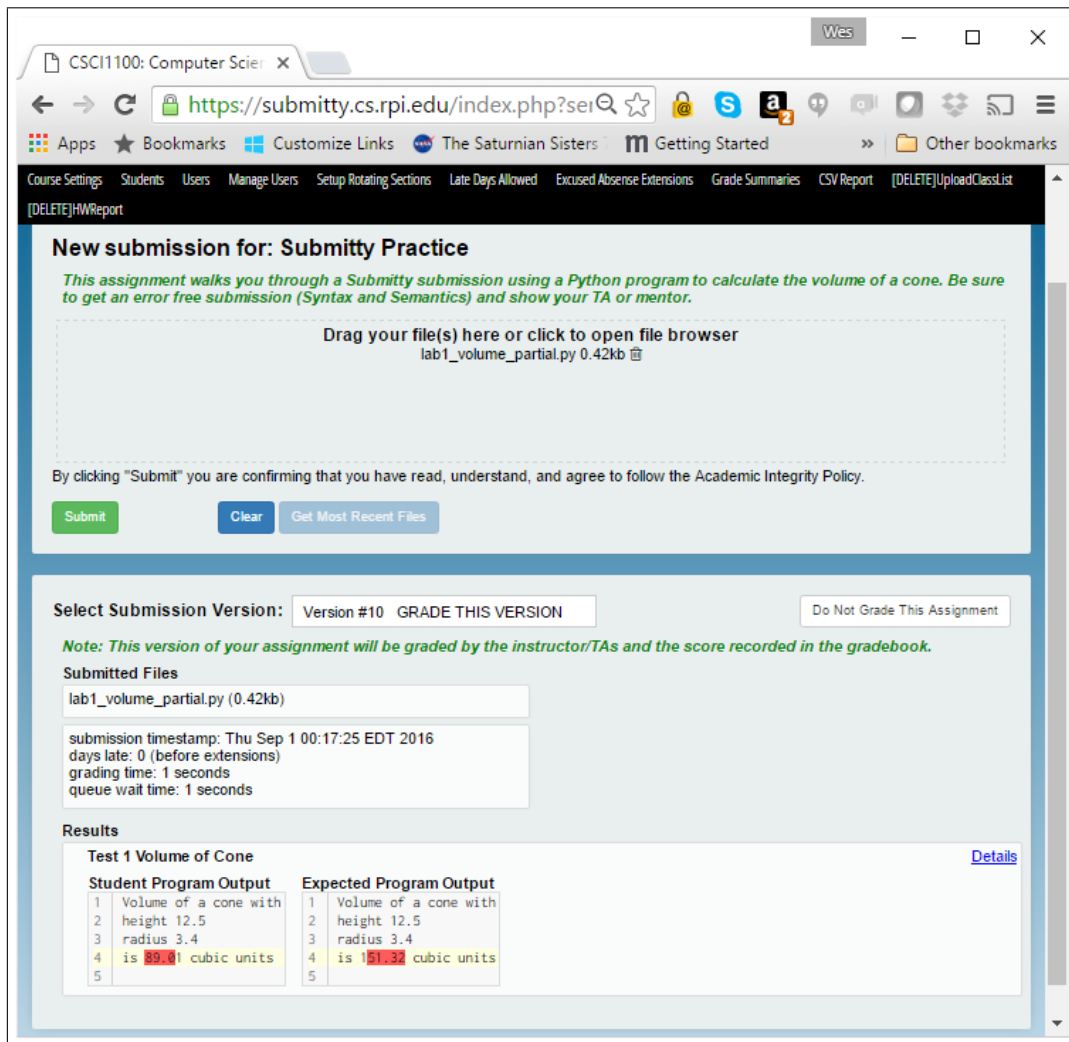
and login with your RCS id and password. Note, if your RCS id is `abcde@rpi.edu`, your login for Submittity will be `abcde` and your password will be your RCS password for that account. The submission server is also linked from the Course Website (<https://www.cs.rpi.edu/academics/courses/fall19/csci1100/index.html>) if you need to find it later.

- (c) Once you are logged in, click on the **Submitty Practice** item. This will bring up a submission window. Either drag your `lab1_volume.py` file into the submission box, or click in the box and select your file from the file browser. Click **Submit** to start the grading process. Once the program runs, click on the **Details** text to the right of the test to see the details of the **Submitty** run. Make sure that you do this every time you submit to make sure you catch any errors.



This is what the server shows you when your program has syntax errors. (Python only shows one syntax error at a time, so if you have multiple syntax errors in your program you will only see the first one.)

- (d) Now it is time to find and fix the syntax error(s). It would be best if you do **not** fix any semantic error(s) you may find yet. We recommend that you edit the file in your Spyder IDE and use Spyder to get the program to run rather than using Submitty as a syntax checker. Keep correcting syntax errors and then checking the program by using the green arrow in Spyder until you have made sure that the program runs on Spyder IDE.
- (e) At this point, we want to demonstrate how Submitty shows semantic errors. Submit the syntactically-correct version of `lab1_volume.py` to the homework server by again loading your program into the submission block. Answer Y if asked if you want to replace the file. Once the file is loaded, hit **Submit**, and review the difference between your results and the expected results for any discrepancies. Here is an example



Places where your solution differs from the expected solution are shown in **red**

- (f) Finally, fix the semantic errors in the program and resubmit. Repeat this as many times as you need to until your answer **exactly** matches ours.

To complete Checkpoint 1: show a TA or a mentor that you have successfully submitted both correct and incorrect versions of the program to Submittity.

Final note on the homework server: You have seen at this point that multiple submissions of the same assignment are accepted on the homework server. There is no limit to the number of times you may submit. (You may start getting a warning about the number of submissions after twenty tries, but this can be safely ignored.) The server keeps track of all submissions. Please be aware, however, that the TAs will only look at the **active version** when grading your homework, which is usually the latest version unless you designate an older version as the active version. You can explicitly change the active version by selecting the version from the dropdown next to "Select Submission Version" and click "Grade this version" button.

Also be aware that the server may run your program slower near deadlines, making you wait to see the results of your program. Make a habit of testing your program before you

submit to avoid unnecessary delays. Note that in terms of deadlines what matters is when you submitted the program, not when it showed you the output.

Checkpoint 2: Submitting Lecture Exercises

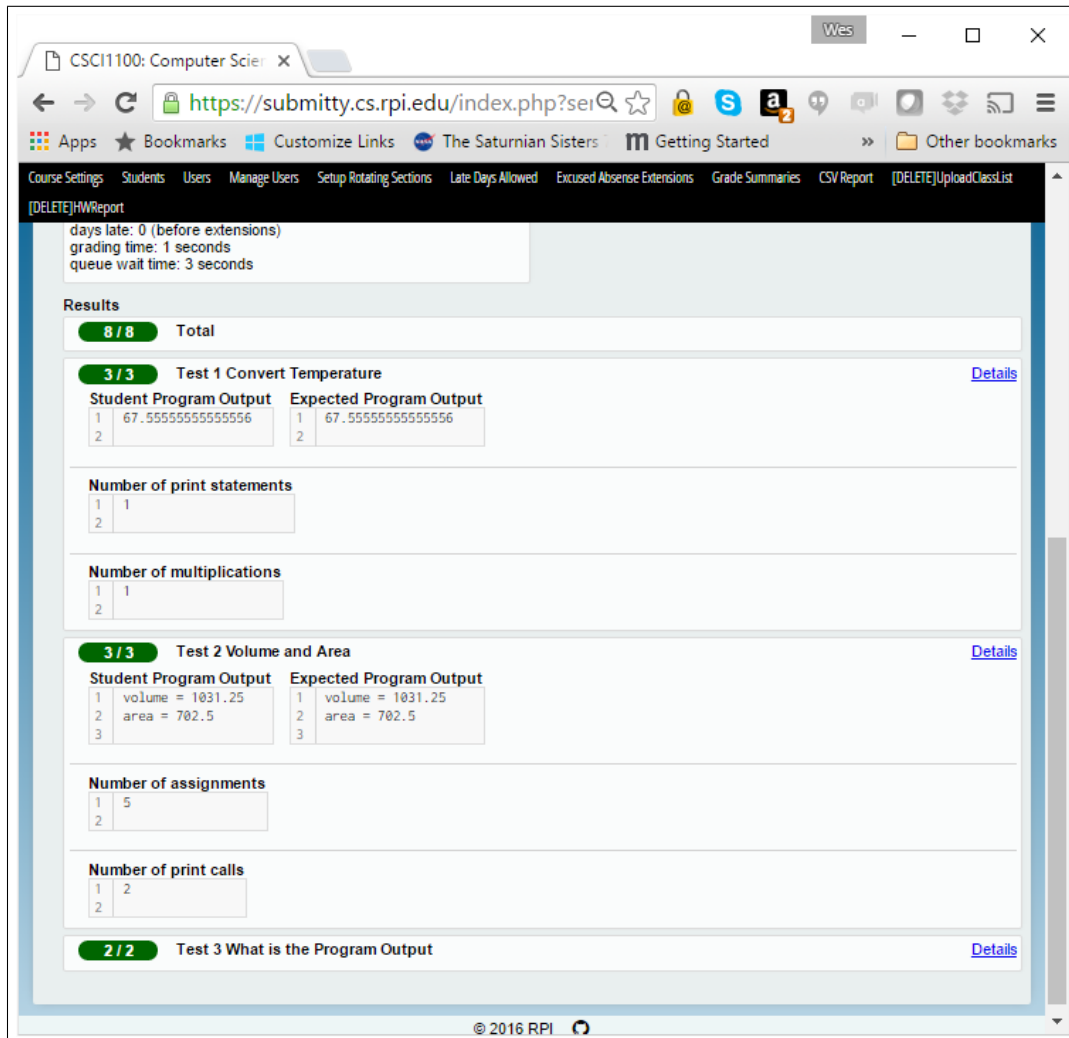
The Lecture 2 exercises asked you to write two short Python programs and to generate a file containing the output from a program. In this Checkpoint you will upload these three files to Submitty.

Start by navigating back to the CS 1 Submitty start page using the links at the top of the page. This time, select **Lecture 2**. Since Lecture 2 asked you to submit two Python programs and one text file, you will notice that there are three submission blocks on the page labeled for the three problems you were asked to solve. These three boxes work independently. You can upload solutions to any or all of the parts independently before running submit. For example, the following screen shows a submission with only the first question answered.

The screenshot shows a web browser window with the URL <https://submitty.cs.rpi.edu/index.php?search>. The page title is "CSCI1100: Computer Science". The navigation bar includes links for Course Settings, Students, Users, Manage Users, Setup Rotating Sections, Late Days Allowed, Excused Absence Extensions, Grade Summaries, CSV Report, and a [DELETE]UploadClassList link. The main content area is titled "New submission for: Lecture 02 Exercises" and includes a warning: "Submit each part of your homework to the right bucket or you will not receive credit." Below this, there are three submission buckets: "Drag your Problem 1: Temperature here or click to open file browser" (with a file named p1_sol.py, 0.02kb), "Drag your Problem 2: Volume here or click to open file browser", and "Drag your Problem 3: Output here or click to open file browser". A confirmation message states: "By clicking 'Submit' you are confirming that you have read, understand, and agree to follow the Academic Integrity Policy." Below this are buttons for Submit, Clear, and Get Most Recent Files. The bottom section shows the "Select Submission Version" dropdown set to "Version #16" with a score of "3 / 8" and a "GRADE THIS VERSION" button. A "Do Not Grade This Assignment" button is also present. A note states: "Note: This version of your assignment will be graded by the instructor/TAs and the score recorded in the gradebook." Under "Submitted Files", the file "part1/p1_sol.py (0.02kb)" is listed. Submission details include: "submission timestamp: Thu Sep 1 01:15:39 EDT 2016", "days late: 0 (before extensions)", "grading time: 1 seconds", and "queue wait time: 1 seconds". The "Results" section shows a total score of "3 / 8" and a breakdown: "Test 1 Convert Temperature" (3 / 3), "Test 2 Volume and Area" (0 / 3), and "Test 3 What is the Program Output" (0 / 2). Each test has a "Details" link.

You can iteratively add the rest of the answers in, and you can change any or all of the solutions independently at any time. When you hit **Submit** the most recent versions of each file will be evaluated and scored. All lecture exercises will be automatically scored

without TA intervention. To make sure that you are following instructions, we have the ability to inspect the code. Clicking on the **Details** link shows what we are looking for in the answer as shown below.



For Lecture 2 we are counting print statements, multiplication operators, assignment statements and plus statements to be sure you are using the correct number based on the instructions. We are being very specific in our messages for this set of exercises; that may not always be the case. If you are having issues getting your tests to pass, **re-read the instructions** and make sure you are following all of the guidelines.

Additional details on using Submittity can be found at <http://submittity.org/student>. Expect them to be updated as the semester proceeds.

To complete Checkpoint 2: show a TA or a mentor that you have successfully submitted correct Lecture 2 exercises to Submittity.

Checkpoint 3: Submitting at least one post on the discussion forum

In this checkpoint, your TA will post a thread on the discussion forum. The Title of the thread will be your Lab Section. Post a brief message under this thread (as a reply) introducing yourself to the class.

To complete **Checkpoint 3:** Show the TA your posted message for this checkpoint.

Checkpoint 4: How big is your hard disk?

Please come to lab for the last checkpoint.