

CS 261 Recitation 1: Setup and C Language

In order to get credit for the recitation, you need to be checked off by the end of recitation. For non-zero recitations, you can earn a maximum of 3 points for recitation work completed outside of recitation time, but you must finish this recitation before the next recitation. For extenuating circumstance, contact your recitation TAs and Instructor.

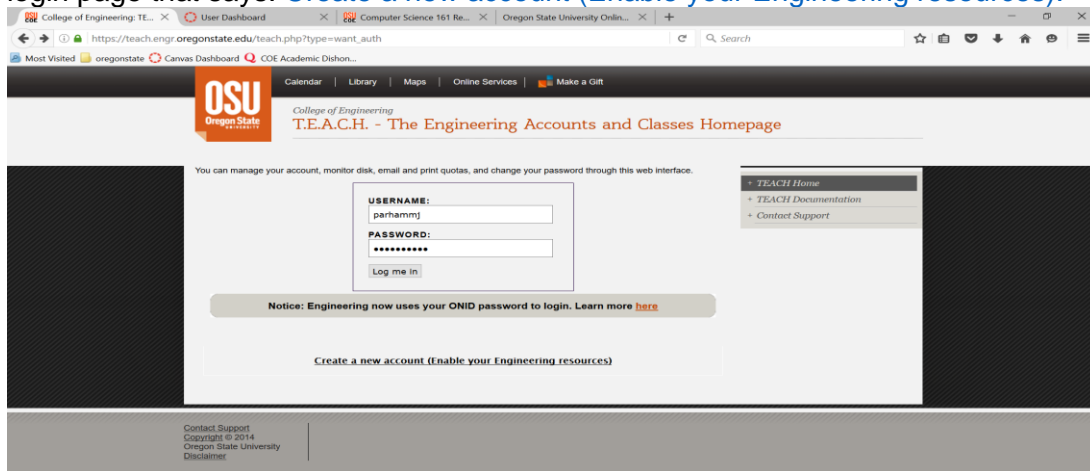
Recitation 1 Grade Breakdown:

- Part 1: Getting Setup 1 pt (Individual work)
- Part 2: C Basics: Conditionals and Loops 2 pts (Group work)
- Part 3: C Basics: Pointers 3 pts (Group work)
- Part 4: Implement Two Simple Programs 4 pts (Individual work)

(1 pt) Part 1: Getting Setup

In part 1 of this week's recitation, you'll make sure you're able to access the College of Engineering servers using SSH to edit, compile, and run code on those servers. Your instructor will help you with this if you're not able. **This is an individual exercise: you'll do it on your own, not in a group.** Here's what you must do:

1. Log into TEACH to see if you have an ENGR account. **If you do not have an ENGR account**, then you will need to create one by clicking on the link at the bottom of the login page that says: [Create a new account \(Enable your Engineering resources\)](#).



2. Use your favorite SSH program (e.g. [PuTTY](#) or [MobaXterm](#) on Windows or the `ssh` terminal command in MacOS or Linux) to log in to the server `flip.engr.oregonstate.edu`.

However you're most comfortable, upload/download the start code for this recitation onto `flip.engr.oregonstate.edu`:

<https://classes.engr.oregonstate.edu/eecs/summer2022/cs261-001/recitations/rec1.zip>

3. There are a few different ways you can do this. One possibility is to download the file onto your laptop and then upload it to `flip.engr.oregonstate.edu`, e.g. using SCP or a program like MobaXterm or [Filezilla](#). Another possibility is to use the `wget` terminal command directly from your SSH connection to `flip.engr.oregonstate.edu`.
4. Unzip the `rec1.zip` using

```
unzip rec1.zip
```
5. Change directory into `rec1`. Either on your local machine or using a terminal-based editor like vim or emacs within an SSH session to `flip.engr.oregonstate.edu`, edit `test.c` to replace the phrase "PUT YOUR NAME HERE" on line 6 with your actual name. If you made this change locally, make sure to upload your edited file to `flip`.
6. Once you have the `test.c` program edited and stored somewhere on `flip.engr.oregonstate.edu`, use the GCC compiler to compile it on `flip` into an executable called `test` (using the C99 standard):

```
gcc --std=c99 test.c -o test
```
7. Run your compiled program:

```
./test
```

That's it for Part 1! Now you know you can compile and run code on the ENGR servers, so you should be all ready to work on the programming assignments for this course.

(2 pts) Part 2: C Basics: conditionals and loops

In part 2, you will be working **in a group** to understand the conditionals and loops in C.

1. Files needed in `rec1` directory: `ex1.c`, `ex2.c`
2. As a group, without executing the programs, trace through the code and determine the purpose of `function()` in each program.
 - a. For `ex1.c`, you may write down the program output in a text file or a document
 - b. For `ex2.c`, you may use a few test inputs, and write down their corresponding outputs in a text file or a document
3. Compile and run both `ex1.c` and `ex2.c`, see if the outputs you wrote down match the actual output.

(3 pts) Part 3: C Basics: pointers

In part 3, you will be working **in a group** to understand the pointers in C.

1. Files needed in rec1 directory: pointers.c
2. Compile and run the program on ENGR server
3. As a group, in a text file or a document, write the output of the six `printf()` in the program
4. As a group, after each `printf` statement, write:
 - a. justification why you got this output and what does this output represents
 - b. the memory addresses of a, b, and c
 - c. the content of a, b, and c

(4 pts) Part 4: Implement Two Simple Programs

In part 4, you will be working **individually** to finish two simple programs.

1. Files needed in rec1 directory: Q1.c, Q2.c
2. Use the instructions in comments to finish the programs.

Make sure you get checked off by the instructor by showing them the output of your program and your group work before the end of your recitation section.

For backup purposes, please submit your work for this recitation (including all documents/text files for group work, and programs) to TEACH.