Homework 1
Welcome, Setup, and Some Light Reading

Due: Wednesday, January 23rd, 11:59PM (Hard Deadline)

Submission Instructions

For all regular homework assignments this semester you'll be using a GitLab repository to submit your homework. You should sign in to GitLab with your UM account. Make sure you are using https://gitlab.umich.edu and **not** https://gitlab.eecs.umich.edu. You'll need to grant the CSP staff (mmdarden, amrith, tarunsk, sltries, samkhan, egajda) reporter permissions to your repo and register it here.

If all of the git stuff is confusing and you don't know what's going on, you can do the HW assignment now and create the git repo later. Lecture 2 covers the basics of git which should be more than enough to get this part set up giving you more than 5 days after lecture 2 get it done. If you're still confused/unsure, feel free to come to office hours.

You need to ensure that your repo follows the following structure. Each week should be its own folder with a homework folder with your markdown file(s). This structure ensures that our autograder can find your homework files. The markdown files for each HW should be the completed templates that are provided for each HW. The markdown files must be named hw[week_number].md. For example, you would have hw1.md for week 1 and hw2.md for week 2. The templates can be found on the course schedule. If you do not use the template(s) provided our autograder will not grade your answers. An example repo for someone who has done the first two HWs is shown below.

```
EECS_201
├─ week01
│  └─ homework
│     └─ hw1.md
└─ week02
   └─ homework
      └─ hw2.md
```

If you were to use this folder for all your CSPrag assignments, your repo might look something like this.

```
EECS_201
├─ week01
│  └─ homework
│     └─ hw1.md
├─ week02
│  ├─ feedback.md
│  ├─ lectureNotes.md
│  ├─ homework
│  │  └─ hw2.md
│  └─ advanced
│     ├─ advNotes.md
│     └─ somescript.sh
└─ week03
   ├─ lectureNotes.md
   ├─ homework
   │  └─ hw3.md
   └─ advanced
      └─ advNotes.md
```

1 Set Up an Ubuntu Virtual Machine

One of the goals of this class is to understand systems work so that you can customize and improve them for yourself. On CAEN, course environments are already set up and everything “just works”. On a brand new Ubuntu install, however, we will have to find, install, set up and manage many tools ourselves.

Recall from lecture that a virtual machine (VM) is a fake computer running as a program. We’ll use a VM in this course as a playground to test things out and work without risking anything on your day-to-day machine. To kick things off, we start by getting a basic environment set up this week.

One final thought: Homework in this class will often be a little underspecified. You are expected to Google, to try things, and to fail from time to time. Making mistakes is highly encouraged, it’s how you learn. We have many office hours if you find yourself getting stuck, but we will always start with the questions, “What have you tried so far?” and “Why do you think that didn’t work?”

1. Get a copy of the **Desktop** version of **Ubuntu 18.04.1** (this is a big download, consider doing it on campus).
2. Download and install VirtualBox.
3. Open VirtualBox and create a new virtual machine. Most of the defaults are fine. The default hard drive size of 8 GB is a little small, I recommend going bigger (50 GB or so). By default, disk images are *sparse*, which means it won’t take 50 GB of real disk space to create a fake disk, rather the fake disk will grow on demand as it’s used, so there’s not a lot of harm in choosing a big number.
4. Install Ubuntu on your new virtual machine. I recommend “Downloading updates while installing”.
5. Once Ubuntu is running, install the Guest Additions (try VirtualBox’s Devices menu → Insert Guest Additions CD Image; you’ll need to reboot once this finishes).
   Q1: What are Guest Additions? What do they do? What changed after you installed them and rebooted your VM?

Guest Additions give the Ubuntu VM some extra features, including automatic resizing, bidirectional keyboard and drag and drop.I noticed it was much easier to resize my VM transfe files from one machine to another.

6. Play around with your new machine! Try writing and running a Hello World program. What about other tools you’ve used before? Can you get an old course project running? How is it different than a CAEN environment?

2 Readings

Each of these are short blog posts, 5-10 minute reads. I selected these to give you a little exposure to some varying perspectives. The authors, Joel in particular, have several other very interesting posts that I highly encourage exploring. After each reading, write a response for the given question.

Biculturalism by Joel Spolsky
http://www.joelonsoftware.com/articles/Biculturalism.html

Q2: Has your computing experience thus far aligned more with "Windows culture" or "unix culture"? What makes you feel that way?

My computing experience has aligned more closely with Unix culture as I frequently use the command line for a variety of tasks, to open documents in sublime or atom or to make my computer sleep. I also feel that I would value a command line or text drive core compared to a GUI.

These two articles use the word "research" a lot, but the points made apply well to any work in computer science.

Helping my students overcome command-line bullshittery by Phillip Guo
http://www.pgbovine.net/command-line-bullshittery.htm
and the counter-point
On the value of command-line "bullshittery" by Eytan Adar
https://medium.com/@eytanadar/on-the-value-of-command-line-bullshittery-94dc19ec8c61

Q3: What did you take away from these articles?

I found both of these articles very relatable, as I know I have given up on projects because I have gotten too stuck on all the command line bullshittery. These articles bring up the very interesting point that it is important to struggle with this in order to develop coding meta cognition and therefore become a better developer. It has encouraged me to allow myself to struggle with the command line in order to be able to improve my problem solving skills.