Name:_________________________________ **Section:**__AY____

Objectives

- Access the course home page.
- Familiarize yourself with the Linux environment.
- Learn how to access and use the Jupyter notebook platform.
- Learn how course policies will be applied with respect to labs.
- Explore basic concepts of computer programming.

Accessing the Course Web Site

1. Press CtrlAltDel at your workstation.

The system will ask for your user name—this is your NetID, or the prefix of your illinois.edu *email address that you've been assigned as a student at the university (but don't enter* @illinois.edu *afterwards). (This is* not *your UIN, or University Identification Number, which you should keep confidential.)*

2. Enter your NetID and press Return Enter. Enter your password and press Return Enter .

The system will log you in. This may take a few minutes if you have not logged in before. When you have logged in, the Desktop will appear.

ⓘ If you are not able to log in, double-check that you are typing your password correctly. If the problem persists, ask your TA for assistance or a guest account this week.

3. Along the top of your screen there is a *taskbar*, similar to the menu bar on Mac OS X and the Start taskbar in Windows. Click on the Applications menu, then the Internet item, and open Firefox Web Browser. In the browser's address bar, enter go.illinois.edu/cs101.

This takes you to the CS 101 home page.

4. Log in to the home page using your NetID and password. There will be a grey link Sign In at the top of the screen; click it to enroll in the course. When prompted for an email address, use your illinois.edu address. Verify the account by checking your email and clicking the link sent in the email with subject line "Your RELATE sign-in link".

5. Click on CS101 Fall 2015, Enroll » (at the top, if prompted to do so).

You are now signed in and can access available course materials.

Accessing the Jupyter Notebook Platform

6. Press CtrlAltT at your workstation, or find *Terminal* in the *Applications* menu.

A Terminal, or command line, will open.

7. Type /class/cs101/startup and press Enter. (Don't forget the / at the beginning.)

The startup script will ask you which section you are in and set things up for you.

8. When the previous step completes, open a new Terminal and type jupyter-notebook.

This opens the Jupyter notebook platform, where your labs will take place. You will often use this interactive notebook platform to compose code and complete exercises.

9. Open up the *Assignments* tab and *Fetch* lab01 for the next part of today's assignment.

ⓘ Illinois Compass (compass2g.illinois.edu) will also be used for some course functions, such as grading and registering your iClicker.

When you are done with the Jupyter assignment, the final part of today's lab exercise is on the back of this sheet. Turn it in to your TA when you have completed it, and also *Submit* lab01 online.

Thinking in Code

The art of programming a computer depends on our ability to clearly express what it is that we want the computer to do. The machine only knows what we tell it to do, and it’s incapable of inferring things we leave out or express imperfectly! Even simple tasks like displaying an image require hundreds or thousands of lines of code, from the machine language of the monitor and its driver up to telling the operating system where and how to position the image. Fortunately, most of this work is already taken care of by high-level languages like Python or Javascript and system libraries.

1) Consider attempting to locate a room in an unknown building. You walk in the front door. What do you do next? Where can you go to try and what do you do when you get there? Write a “recipe” explaining what process (algorithm) you follow as you locate the room. Your response should include at least 3 steps—we don’t expect it to be exhaustive but it should cover major cases.

This sort of explanation is called *pseudocode*, which forces us to write out all of the steps so that we understand how to write the real program later.

2) A common engineering solution method is the rough estimate (or “back of the envelope” calculation). How would you estimate the number of times you’ll press the Return Enter key in your lifetime? Write out a reasonable process—as steps or an equation—with at least three components.

3) Programming often relies on conventions (for example, applying order of operations) to express things. For instance, consider the mathematical expression:

$$5x + 7x - 12 = x + 4$$

which you can simplify by the rules of algebra to solve for x. Solve for x □□□ □□□□ □□□□□ □□□□ (there should be at least three)!

This forces us to think about the *process* explicitly, rather than automatically (or by skipping steps). Programming requires us to be explicit and complete in directing the machine, and tends to be unforgiving of muddy thinking or imprecise expression.