

Name: _____ Student ID: _____ Section: _____



This file is located on

<https://relate.cs.illinois.edu/course/zuics101fa16/f/labs/lab01.pdf>

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. Use a browser (e.g., Firefox, IE) to visit the course website <https://relate.cs.illinois.edu/course/zuics101fa16/>.
2. When you visit <https://relate.cs.illinois.edu/course/zuics101fa16/>, please make sure that you are logged in in order to conduct your homework assignments, etc. If you see "You're not currently signed in. Sign in >>" near the top of the course webpage, then you know you haven't signed in yet. You can click "Sign in >>" button near the top of the course main page, and then click "Sign in using your email >>", ... following the steps in the second slide of the lecture 1 slides: <https://relate.cs.illinois.edu/course/zuics101fa16/f/lectures/lec01.pdf> except that you don't need to do Step 6 "Enroll" (which is needed only before you enroll in the course).

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

Accessing the Jupyter Notebook Platform

3. Log in to a Lab machine (with Windows 10 installed).
4. Open a File Manager to navigate to check which user directory your login has as the root directory of the Jupyter Notebook, in short as **the Jupyter Notebook root directory**. Typically it is located as C:\用户\TNNNN\我的文档 (i.e., C:\Users\TNNNN\Documents) where NNNN are digits.
5. Download the following two files and save them under the Jupyter Notebook root directory:
 - <https://relate.cs.illinois.edu/course/zuics101fa16/f/labs/lab01.ipynb>
 - <https://relate.cs.illinois.edu/course/zuics101fa16/f/labs/rocket.py>
6. Double click the "Jupyter Notebook" icon on the desktop of the Lab machine to run the Jupyter Notebook.
7. In the Jupyter Notebook main page (being automatically opened in a browser such as Firefox), click the "lab01.ipynb" file under the list of files/directories to open the Lab 1 Jupyter Notebook file.
8. Follow the instructions listed on the Lab 1 Jupyter Notebook file to conduct Lab 1.



The Jupyter notebook platform is where your labs will take place. You will often use this interactive notebook platform to compose code and complete exercises..

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.

Name: _____ Student ID: _____ Section: _____

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 by using a number of rules of algebra (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.