Name:_____ **Student ID:** _____ **Section:** _____

This file is located on
http://relate.intl.zju.edu.cn:8000/course/zjui-cs101-fa17/*f/labs/lab01.pdf*
Objectives
- Access the course home page.
- Learn how to access the Codelab environment (optional)
- Learn how course policies will be applied with respect to labs.
- Explore basic concepts of computer programming.

Due to special conditions, please work on your labtop and connect to campus wifi first.

Accessing the Course Web Site
1. Use a browser (e.g., chrome, firefox, IE) to visit the course website
   http://relate.intl.zju.edu.cn:8000/course/zjui-cs101-fa17/
2. When you visit http://relate.intl.zju.edu.cn:8000/course/zjui-cs101-fa17/, please make sure that you are logged in first, 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 third slide of the lecture 1 slides:
   http://relate.intl.zju.edu.cn:8000/course/zjui-cs101-fa17/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 have access to all the available course materials.

Accessing the CodeLab environment
1. Use your browser to visit the CodeLab website:
   http://www.tcgo1.com/
2. For the first time usage, you need to register an account with the Codelab service. Warning: Register using your university email address!
3. Register for Codelab:
   a) Click "register for CodeLab"
   b) Enter your university email address, and your real first and last names. Verify and submit
   c) After you receive the verification email, go ahead and complete your registration.
   d) Next, you need to enroll in the CS 101 course. When you are logged in, click "Add a course" at the upper left.
   e) Select "I am in a course that uses CodeLab", and enter the section access code "***ZUIA-25895-NDBB-34***"; continue.
   f) Accept the license agreement and finish any forms.
4. After adding the course, click the "Load" button to load the course (CourseNNN #51977). You should be able to find an "**exercise1**" subtree under the CodeLab for CS101 session in the left "Table of Contents" window.
5. We have prepared 8 python exercises for beginners. Read the instructions of each exercise and try them out. You may refer to the "solutions" in the Workbench window or ask TA for help.

(i) CodeLab is the platform for the first part of our homework assignments and submissions. The exercises in this part is not graded. But it helps you get familiar with the process.

When you are done with the above tasks, the final part of today's lab session is on the next page. Turn the 2^(nd) page to the 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 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.