

# CSC120H Lab 1

Congratulations on getting started with the CSC120 labs. Now it's time for you to do some programming! To earn your 1% lab mark, you must work hard with a partner to finish the lab exercise. If you finish before the end of the lab, show your TA your work. Make sure your TA has marked you as present in the lab before you leave.

In this lab, you will be setting up your Teaching Labs account if you haven't done so yet, and using the Python shell to run some basic code. If you have time at the end of the lab, you can get help installing Python and Wing 101 on your own computer if you need assistance.

## 1 Objectives

1. Learn how to program in pairs as *driver* and *navigator*
2. Define *driver* and *navigator*
3. Practise working with expressions in the Python shell
4. Practise working with variables
5. Use some mathematical functions

## 2 Driver and navigator

You are **required** to find a lab partner to work with and to work in a group of two, using one computer together. We encourage pair programming as it helps when you are stuck. We will use the terms *driver* and *navigator*. Here are the definitions of the two roles:

**driver:** The person typing at the keyboard.

**navigator:** The person watching for mistakes, and thinking ahead.

Here is the most important rule for this and all future labs:

**The navigator must not touch the keyboard or mouse.** The driver's role is to work with the computer, and the navigator's is to think about what is happening on the screen, and about the upcoming issues related to the problem being solved. The navigator can tell the driver what they're thinking and if there's something the driver should notice or has missed.

In every lab handout, we'll call you two **s1** and **s2**, and **s1** will be the first driver, and you will switch as you go along.

Although you can bring your laptop do the lab on it, **we recommend doing your work on the lab computers**, so that both partners can see everything on the big screen.

## 3 Logging In

Whether or not you want to use your own computer to do your work, you need to activate your Teaching Labs account if you haven't done so yet.

Both you and your partner need to activate your account, so take some time before you continue with the lab to do so by logging into your lab computer using the credentials Username: **account**, Password: **account**. Follow the instructions to set up your account.

## 4 Opening Wing in the Teaching Labs

After you've logged in (with either partner's account), you can work on the Teaching Labs computers. To open Wing on the lab machines, please follow the instructions in the screenshots that can be found on the course website:

<http://www.teach.cs.toronto.edu/~csc120h/fall/>

Check the 'Lectures' page for this week and click on 'Wing101 screenshots'. They can also be found on the 'Labs' page.

Follow the instructions there to open Wing.

**At the end of the lab**, you may have time to get help installing Python and Wing on your computer. For now, work on the teaching lab computers with your partner.

## 5 Numerical expressions in the Python shell

Now that the administrative details are out of the way, we can get started with the fun part of the lab! Many of you will be programming for the first time. Don't hesitate to ask your TA or your labmates in the room for help. You may also refer to your notes and the lectures page of the course website.

To begin, make sure you have opened Wing (refer to the screenshots in section 4 if you haven't done so yet). For this lab, we'll start by working in the Python shell.

**Choose your driver/navigator roles (explained on page 1) – one student is s1 and the other is s2.**

**s1 drives and s2 navigates. You will switch later.**

Python has many calculator-like features. Try evaluating the following expressions in the shell.

**Before** you write the expressions in the Python shell, first write your guess as to what value Python will give you for each expression in the 'Your Value Guess' section. Then write the expression in the Python shell and note down the actual value. If there are differences between your guess and the actual value, think about why that is the case. **s1 (the driver) will write the expressions into the shell.**

Expression	Your Value Guess	Actual Value in Python shell
$34 + 8$		
$18 - 2.9$		
$29 / 3$		
$24 \% 5$		
$5.5 * 4$		
$27 \% 3$		
$28 // 3$		
$4 * 3 + 2$		
$4 * (3 + 2)$		
$30 // 3$		

**Switch roles: s2 is now the driver and writes into the Python shell.**

Use the shell to find the result of each of the following. s1 should fill in the table as s2 drives.

Description	Python Expression	Result
subtract 543 from 1234	1234 - 543	
divide 100 by 6		
find number of full 3-person groups we can make with 250 students	250 // 3	
find number of students from above who are left without a group		
64 cubed	64 ** 3 or pow(64, 3)	

In the shell, create variables x and y and z assigning them the values 5, 8 and 0 respectively. Use the shell to evaluate the following expressions and complete the next table.

Python Expression	Result	Type of Result
x + y	13	int
min(x, y, z)		
abs(x)		
y / x		
x // y		
z / y		

Try the expression y / z. Discuss with your partner what happens and why.

**Switch roles: s1 is now the driver and writes into the Python shell.**

Python allows you to **convert** between different numeric types.

For example, if you give it the command `int(3.7)` you get back 3 (try it). This converted the float 3.7 to the next *lowest* integer.

What happens when you enter `float(3)` into the shell? How about `int(3.0)`? Explain to your partner why you get the result Python gave you.

Expression	Your Value Guess	Actual Value in Python shell
<code>28 / 6</code>		
<code>int(28 / 6)</code>		
<code>float(28 / 6)</code>		
<code>type(28 / 6)</code>		
<code>type(int(28 / 6))</code>		
<code>type(float(28 / 6))</code>		

Do the same for these rounding examples:

Expression	Your Value Guess	Actual Value in Python shell
<code>round(14.2)</code>		
<code>round(13)</code>		
<code>round(7.324, 2)</code>		
<code>round(7.324, 3)</code>		
<code>round(264.231, -2)</code>		
<code>round(52.5)</code>		
<code>round(52.5, -1)</code>		

## 6 Variables

**Switch roles: s2 is now the driver s1 is the navigator.**

The values in the expressions that we are working with often have more meaning than the numbers alone reveal. For example, the expression `0.5 * 10 * 5`, might be the area of a triangle, where the value 10 is the base and 5 is the height. To associate a name with those values, we use variables.

The general form of an assignment statement is: `variable = expression`

Perform the following steps in the Python shell.

**Switch roles: s1 drives and s2 navigates.**

- Assign the value 10 to a variable named `base` and the value 5 to a variable `height`.
- Now rewrite the expression `0.5 * 10 * 5` so that it uses variables `base` and `height`.
- Next, assign the expression to a variable named `area`.
- Reassign variables `base` and `height` the values 8 and 4, respectively. What is the value of `area` now? Type `area` and the return key to see `area`'s value. If the value of `area` is not what you expected, and you can't figure out why, call your TA over for help.
- Verify the current variable values by typing each variable in the shell followed by the return key. Now recalculate the area (using the variables) and assign it to the `area` variable. Check that its value is correct.
- *Python shell Tip: Try using the up/down arrows of your keyboard. See what happens if you press any one of them multiple times.*

## 7 Variable Expressions

In the shell create a variable named `navigator_age` and assign to it the navigator's age. Create another variable `driver_age` that is assigned (unsurprisingly) the age of the driver. Now use these variables in Python expressions in the shell to complete the table below.

Description	Python Expression	Result
the driver's age	<code>driver_age</code>	
the navigator's age 12 years from now		
the age of older partner		
the difference between the ages		

Switch roles one last time: `s1` is now the driver `s2` is the navigator.

## 8 Assignment statements

Each line of code below is an assignment statement. Beside is line, there is a space for you to fill in your prediction for the current value of each variable. Since we know that the comment lines (the ones with `#`'s) don't affect the program behaviour (comments never change the values of any variables), we have put them together with the next line of code. We have completed the first two rows of the table for you. Notice that the second line of code doesn't change the value of `completed`. The table should end up full.

**Don't immediately type each line into the shell.** Complete the table with your partner. Only once you're confident in your answer should you try checking the value of the variables at each step.

Assignment statements	completed	required	credits_to_go
<code># when you arrive at UofT you need 20 credits completed = 0</code>	0	—	—
<code>required = 20</code>	0	20	—
<code>credits_to_go = required - completed</code>			
<code># but by Christmas break, you've done 5 courses completed = 5</code>			
<code># oops. They were 5 half-courses completed = completed / 2</code>			
<code># let's update the credits left to take credits_to_go = required - completed</code>			
<code># now by April, you have completed 5 full credits completed = 5</code>			
<code># by the end of 2nd year, you have done 5 more completed = completed + 5</code>			

## 9 Setting up Wing on your own computer

If you managed to install Python and Wing on your personal computer/laptop, you can skip this section. Otherwise, if you brought your laptop with you, you can follow the directions on the ‘**Software**’ page of the course website. If you have any issues during this process, ask your TA (or other students around you in the class) for assistance.

When you are done the lab, show your TA your work. Make sure you are marked present on the attendance sheet.

If you did not finish, or you have questions, finish the lab at home this week, and get any questions answered in office hours or on the discussion board.