CS 115 Lab 0: An Introduction to IDLE and CS 115

Throughout this course, we will be learning how to use Python and programming in general to solve various problems.

But in order to be able to do that, we must first know how to write those progams. And not only in the sense of “learning the language”, but also in the sense of “learning the tool”. Much like how learning to write in real life involves learning both the alphabet and how to wield a pencil, learning to program requires learning software to use to write.

Thus, this lab is designed to introduce you to IDLE, the editor we will be using throughout the semester. Additionally, this lab will also introduce to a number of additional concepts relevant to this class, which you will encounter again during this course:

- How to submit your code through Gradescope.
- How to receive your grade through Gradescope.
- How to properly format your answers, both in Python and outside of it.

Task 1: Installing Python

For the first step, we want to install Python, which comes with IDLE built in.

Before you go downloading it, however, you should **check whether your already have Python installed**. Many modern day computers already have a version of Python installed on them by default. Before you go about following the installation steps, first check if your computer already has Python on it, and whether you can open IDLE. If so, and if the version of Python you have is Python 3, you can skip the rest of this task.

To download Python, go to `https://www.python.org/downloads/`, and click the button to download the latest source release (there should be a large button for it at the top). After downloading, follow the installation instructions listed there (these will depend on your OS).

CA Check-In

Throughout the semester, you will be expected to check-in with your CA during lab, to show that you are making progress. Today’s check-in will be basic, but it should still introduce you to the pattern of talking to the CA.

After you complete the above task, talk to the CA in your section. Show them that you have IDLE open. In particular, show them the version of it you have, to ensure that you have the right version of Python installed.

Then, run the following command, and verify that you get the following output.

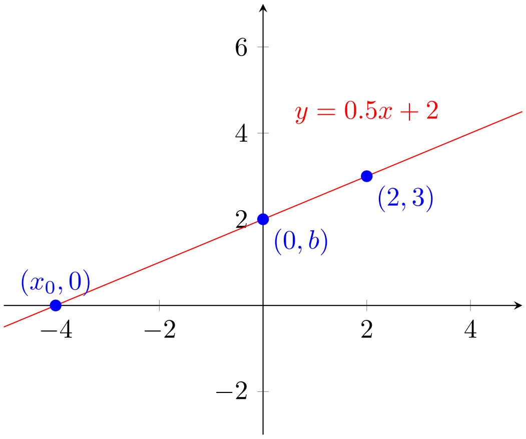
```
>>> print("Hello", "World!")
Hello World!
```

Show that output to the CA to conclude your check-in.

Task 2: Simple Arithmetic

In every lab and homework, we will ask you to implement some functionality. For this lab, we want to give you a very simple function to reason about.

Consider a linear function, of the form $y = mx + b$. Here, m is the slope (how much the function grows as x increases), and b is the y-intercept (the value of the function at $x = 0$).



Given such a function, it is helpful to know its x-intercept (at what x value the function becomes 0). This is easy to calculate using the simple formula

$$x_0 = -\frac{b}{m}.$$

Your job is to implement the function `find_zero`. This function takes in two arguments m and b, and returns the x-intercept of the function with that m and b. For example, the function $y = 5x + 10$ has an x-intercept of -2, so we should see the following behaviour with your `find_zero` function:

```
>>> find_zero(5, 10)
-2.0
```

However, note that when $m = 0$, then the function doesn't have a well-defined zero[1]. As such, in this circumstance, your function should instead return `None`.

Task 3: Running Tests

In the labs and homeworks, we will often ask you to write some code. But how do you know that it is correct?

To help with that, on (most) labs and homeworks, we will provide you with a test script. For this lab, this is the file `test_lab0.py`.

We are not going to cover the details of how this works for now (but this will be explained in lecture in a few weeks). However, we will explain how to run these tests, to verify the correct behaviour.

First, ensure that the test file is in the same directory as your `lab0.py` solution. You can then run this file the same way you would run any Python file in IDLE. However, if you run the test file, you will see that it will run several tests, and tell you whether they passed or not.

If you see something like the following, then you know that your code passes all the tests, and is likely to be correct:

```
....
----------------------------------------------------------------------
Ran 4 tests in 0.000s

OK
```

Otherwise, you will end up seeing several errors or test failures, getting code like the following:

```
.FF.
======================================================================
FAIL: test_find_zero_case_2 (__main__.TestCases)
----------------------------------------------------------------------
Traceback (most recent call last):
```

[1] More specifically, it either doesn't have a zero, or it is zero everywhere. Either way, there is not a unique x-intercept.

```
  File "some/long/path/test_lab0.py", line 9, in test_find_zero_case_2
    self.assertEqual(10.0, lab0.find_zero(0.5, -5))
AssertionError: 10.0 != 20.0

======================================================================
FAIL: test_find_zero_case_3 (__main__.TestCases)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "some/long/path/test_lab0.py", line 12, in test_find_zero_case_3
    self.assertEqual(0.5, lab0.find_zero(-4, 2))
AssertionError: 0.5 != -0.125

----------------------------------------------------------------------
Ran 4 tests in 0.001s

FAILED (failures=2)
```

In that case, you should go back to figure out why your code is not working correctly.

Before submitting your work for any lab or homework, make sure that it passes all the test cases we have provided you.

Task 4: Getting Familiar with IDLE

On every lab and homework, we will provide you with a `report.txt` file. This file is used for you to make comments on your development, but will also sometimes be a place for us to ask additional questions for that lab or homework.

On every report, we provide you a place to state your name, write the pledge, and list your collaborators and resources used. We will also ask you for any problems you ran into during development, and how you figured out how to work around them.

For this lab, we also have one lab-specific question, concerning the use of IDLE. Please ensure that you answer that question before submitting.

Rubric

Category	**Description**	**Points**
Name	For stating your name on every file for submission.	10
Pledge	For writing the full Stevens Honor Pledge on every file for submission.	10
Check-In	For performing the CA Check-In during the lab.	20
`find_zero`	For correctly implementing `find_zero`.	40
Report Questions	For filling out the questions on the report.	20
Total		100

Submission Instructions

For this lab, you must submit the following files, with exactly the following names:

- `lab0.py`
- `report.txt`

Remember to put your name and the Stevens Honor Pledge on every file you submit! Failure to put the pledge can result in marks being deducted, with repeat penalties potentially getting stricter between assignments.

If your code does not compile (throws a `SyntaxError` on being run), you will receive a 0 for the assignment! Make sure you submit code that can be run, even if it does not pass all the test cases.

Submissions must be handled through Gradescope (accessible through the assignemtn page on Canvas). Grades will be released through Gradescope once the lab is graded.

If you are having trouble submitting to Gradescope, please let us (the CAs and professor) know, so that we can assist you with a timely submission.