CS 111 Law
Program Design I (Python 3)
Fall 2019

Lab #02: Basic Python Programming

Complete By: Thursday, September 5th @ 11:30 pm
Assignment: zybook “Lab 2” (two sections late in Chapt. 2)
Policy: Individual work only, late work *is not* accepted
Submission: .py files submitted via zybook sections.

Installing software

This resources page of the course website tells you how to install the programs we'll be using on your machine. Everything you'll need comes just from downloading Anaconda's distribution of Python 3, which gives you Spyder. Spyder is an Integrated Development Environment (IDE) that contains both an editor for Python code and a *console* (i.e., a place to execute Python). Technically it is an iPython console (meaning it will play nicely with images such as the graphs we will plot later).

Trying out Python at the iPython console

First start the Anaconda Navigator, and then from it, choose Launch Spyder.
(You may see an invitation to update to a slightly more recent version of Spyder, e.g., from 3.2.5 to 3.2.6, which you can ignore if you want, or you can do that.)

In the right-hand window with title “Console <something>” you should soon see a prompt that looks like
In [1]:

Note: you may see some messages first; e.g., you might see:

```
Python 3.7.0 (default, Jun 28 2018, 07:39:16)
Type "copyright", "credits" or "license" for more information.

IPython 7.6.1 -- An enhanced Interactive Python..
```

- In general, this “console” is an area for experimenting with the Python language. The “prompt” tells you that Python is ready to go. If you liked the episode "42" of Dr. Who Season 3, you might try 6*7 as a first computation at the prompt.

- Now, try some larger computations... Try computing a googol (*yes, that is spelled correctly*—it's ten to the hundredth power). The power operator in Python is two asterisks **. So, at the prompt you would type
    - `10**100`
- Next type or paste this line of Python code:
    - `print("The best class in CS is ", 4+25*4+7)`
    - You should see the output: `The best class in CS is  111`
- And experiment a bit more on your own for a few minutes.

Assignment: Using Python with Spyder's editor

The assignment is to solve two additional programming exercises outlined in the CS 111 zybook, near the end of Chapter 2. Specifically:

- Hello from Spyder (zyBook 2.13)
- Hello from somebody we specify (zyBook 2.14)

(You may need to refresh your browser to see these sections.)

[*Note: the sections were added to the end of Chapter 2 because we wrote them.*]

Have a question? Use Piazza, not email

As discussed in the syllabus, questions must be posted to our course Piazza site — questions via email will be ignored. Remember the guidelines for using Piazza:

1. *Look before you post* — the main advantage of Piazza is that common questions are already answered, so search for an existing answer before you post a question. Posts are categorized to help you search, e.g. "Lab2".
2. Post publicly — only post privately when asked by the staff, or when it's absolutely necessary (e.g., the question is of a personal nature). Private posts defeat the purpose of piazza, which is answering questions to the benefit of everyone.
3. Don't post your entire answer — if you do, you just gave away the answer to the ENTIRE CLASS. Sometimes you will need to post code when asking a question --- in that case post only the fragment that denotes your question, and omit whatever details you can. If you must post the entire code (e.g., when asked to do so by one of the staff), then do so privately --- there's an option to create a private post ("visible to staff only").
4. Ask pointed questions — do not post a big chunk of code and then ask "help, please fix this". Staff and other students are willing to help, but we aren't going to type in that chunk of code to find the error. You need to narrow down the problem, and ask a pointed question, e.g. "on the 3rd line I get this error, I don't understand what that means…".
5. Post a screenshot — sometimes a picture captures the essence of your question better than text. Piazza allows the posting of images, so don't hesitate to take a screenshot and post; see http://www.take-a-screenshot.org/ for quick ways to take a screenshot on your platform of choice.

Submission and Grading

Your answers are automatically submitted via your zybook as part of the assigned exercises / sections. The grading scheme at this point is based entirely on correctness --- i.e., the score that zyante reports. If the project involves multiple exercises, we will generally give the weights for each. (Otherwise, they're equally weighted.)

Policy

Late work ***is not accepted*** for labs.

All *coding submitted for a grade* is to be done individually. You are welcome, even encouraged to discuss solutions with others, but you must leave group meetings with no notes, and then write your own code.

The University's policy is described here: https://dos.uic.edu/wp-content/uploads/sites/262/2018/10/DOS-Student-Disciplinary-Policy-2018-2019-FINAL.pdf

In particular, note that you are guilty of academic dishonesty if you extend or receive any kind of unauthorized assistance. Absolutely no transfer of program code between students is permitted (paper or electronic), and you may not solicit code from family, friends, or online forums. Other examples of academic dishonesty include emailing your program to another student, copying-pasting code from the internet, working in a group on a homework assignment, and allowing a tutor, TA, or another individual to write an answer for you. Academic dishonesty is unacceptable, and penalties range from a zero in a lab to failure in the course to expulsion from the university; cases are handled via the official student conduct process described at https://dos.uic.edu/community-standards/academic-integrity/ .