Introduction to Programming for the Life Sciences
Fall 2021

Instructor: Seth Syberg
Contact: ssyberg@rockefeller.edu

Teaching Assistant: David Requena Anicama
Contact: drequenaan@mail.rockefeller.edu

Meeting Times: 10 virtual meetings starting September 15th, 5pm – 7pm
Office Hours: Weekly TBD

Approximate Time/Communication Commitment: 8 to 10 hours / week. Contrary to the popular belief programming is hard! I highly suggest only taking this course if you have the time to commit to it. It's very easy to fall behind and very difficult to catch up.

Additionally, ***you are expected to have frequent (multiple times/day) access to email, the course website, and the course online forum***. As with the nature of programming the course is always evolving, bugs in the labs and assignments are being found and solved, hints and changes are common, so staying abreast of updates and announcements are crucial for success.

Computer Requirements: As this is an online course with a live video lecture, a computer with a strong internet connection is required. It's also strongly suggested you have a relatively new / well-functioning computer. Some of the work we do in this class is processor intensive and we will be using recent versions of software that may require recent versions of your operating system. Rule of thumb, if your computer is *more than 4 or 5 years old* you may struggle in this course. Most instructions will be given assuming Windows, but you should be fine using a recent version of an Apple OS as well.

Acknowledgements: A HUGE thank you to my dear friend Dr. Josh Hug (the instructor for this course in 2013 and 2014) for providing us with his course materials that can be found, modified by myself, throughout this site!

Course Blurb. Here's a neat fact you may not know: Once you know how to write code in a single programming language, you can automate any task in the universe that is possible to automate (including perhaps even conscious thought, though that level of program will be beyond the scope of our course).

You may already perform such tasks manually, or perhaps with the assistance of other people's special purpose tools that might not do exactly what you want. These tasks include things like massaging data in spreadsheets into a more useful format, making

data plots, or searching for interesting patterns in data (sequence alignment, cell identification in microscopy images, etc.).

In this course, we'll learn how to build tools to perform such tasks from scratch, as well as how to modify existing open source tools. This course will focus primarily on learning to write general purpose programs in the Python programming language.

Prerequisites. I'll be assuming no prior programming knowledge.

For those of you with some programming background, you may skip the first few classes but please run it by me first (though you still must complete all labs and assignments). All materials will be posted online.

Course Format. The course will consist of the following main components:

1. **One weekly two-hour meeting:** I attempt to minimize lecturing and focus more on live coding/demos and hands-on programming exercises.
    a. **Weekly office hours (1-2 hours):** For those who need additional personalized guidance.
2. **On going programming challenges (labs & assignments):** To be completed outside of class, approximately 8 to 10 hours / week.
    a. **Labs:** One or more sets of practice questions per week, reinforcing and extending concepts from lectures.
    b. **Assignments:** Larger questions requiring one to synthesize, apply, and extend concepts from lectures and labs.
3. **Online discussion forums**: For assistance with challenges and general questions we will make extensive use of an online forum (and perhaps a “chat room”) where students will post questions, help fellow students, and engage in discussion.
4. **Final Assignment**: You will create a project that is something you can use in your own research, utilizing nearly every major concept learned in class. Due at the end of the course.

Lecture Times. Class will meet from 5pm – 7pm for 10 sessions between September 15th and December 15th*.

**There will be a mix of consecutive weekly meetings and skipping weeks to accommodate the pace and difficulty of different sections of the course.*

Weekly office hours led by Teaching Assistant:
For those who need a more personalized guidance, they can attend office hours with the TA (dates/times TBD).

Textbooks. There are no required books for this course, though we will make use of many online references which will be posted in forums, labs, and assignments throughout the course.

Course Website: For this class, we will be relying very heavily on the class website offered through the Coursemology platform. This is where all materials, labs, assignments, and announcements will be posted, and where many labs and assignments will be completed.

Course Forums and Discussion Rooms. Online forums are an invaluable resource for any programmer. As such, we will make extensive use of a course forum and discussion room hosted by Campuswire. Most questions in this course should be directed to the forum where they will be answered by me or your fellow classmates. More often than not, a given student's question is one I get from multiple students, and the answer may prove useful to everyone.

If you email me directly I will often refer you to Campuswire before answering your question (if I deem it appropriate).

Do not, however, post your solutions (or partial solutions) for any assignments to the forum *in answer to another student's question*. To avoid posting solutions to the forum you may produce an unrelated example that exemplifies your answer for safely posting in the forum.

> Campuswire Guidelines:
> Before posting to the forum - please think about what me and your fellow students will need to help you! For example:
> - If you are posting about a failing test (test suite or Coursemology) always include any/all output you are getting
> - If you are getting a python error - post the error
> - If you are having a file system issue - post a screen shot
> - The more information the better!

Final Project. Create a project that is something you can use in your research utilizing nearly every major concept learned in class. If you're unable to come up with something for research, then something that is directly in your field of study/specialty.

Components: (***No late work will be accepted!)***

1. Students will first submit a proposal detailing the problem or set of problems they'd like to solve. At minimum it should include all program inputs, outputs, and a summary/description of the project.
2. The completed project will include your project code, as well as test client code, and a video verifying the performance of the project code.

3. A short report will be submitted with the completed project, describing the current state of your project, the outputs of the program, and overall conclusions.

Minimum Requirements:
Your project must be substantial (i.e. hard). 30 lines of code that meet all the technical requirements *isn't going to cut it.*

Your project must also make use of nearly every major construct learned throughout the course:

- Loops
- Conditionals
- Advanced data structures (lists, dictionaries, classes)
- Plotting or an equivalent visual component
- External files / external data of some kind

Collaboration Policy. Students are welcome to collaborate on labs and assignments, with the exception of the final project. If collaborating please note who you are collaborating with in the code/lab/assignment (generally as a code comment).

Grading. This is a pass/fail course, *but it's an easy course to fail*. A passing grade on **all** labs and assignments is mandatory to pass the course. Any late submissions require prior approval from me and will never be accepted after the solution is posted – these submissions will result in a zero. If you get a zero on any lab or assignment – or think you might miss a deadline -- *please notify me asap to discuss next steps.*

Most assignments will be accompanied by a test suite or Coursemology tests. In order to receive a passing grade on any assignment, *you must pass all automated tests.*

The very best way to avoid failing the course is to get help early and often!

Satisfactory completion of a final project is also required for passing the course. Activity in the forums will be viewed favorably.

Visualizer. Python Visualizer available here:
http://www.pythontutor.com/visualize.html#mode=edit

Software. We will also be making extensive use of the PyCharm IDE (integrated development environments) to complete all coursework. Additionally, we will use Coursemology's online code testing for many labs and assignments.

Instructor Bio: Seth Syberg has been a computer science and information systems lecturer for a number of years, working at many community colleges and universities, most recently at the University of Washington in Tacoma. His breadth of teaching spans

from introductory programming up through graduate level computer science courses. In addition to his work as an instructor, Seth has over 15 years of experience working in the tech industry. He completed a Master's of Computer Science at Brown University in 2007 and began teaching at Rockefeller University in Fall 2015.