BIOL 4750-001/6750-004: Programming for biologists
Fall 2017

Instructor: William D. Pearse

1 Course description and learning objectives

Every modern biologist must be able to *use* computer software, but (sadly) few biologists can *create* software to answer their own novel questions. This course will teach you how to think and act like a computer scientist, quickly writing effective code to get publication-quality results. We will focus on foundational computer science concepts such as algorithm design, making use of modern tools such as GitHub from day one. This emphasis on the fundamentals of computer science and not superficial aspects of programming languages will allow us to cover two programming languages in depth (`R` and `Python`). You will work on a longer-term project simulating an ecosystem ('R-World'), giving you practical experience for real projects once the course is complete.

Specifically, by the end of this course you will:

- Have an advanced understanding of the R programming language
- Have an intermediate understanding of the Python programming language
- Know how to interpret, choose, and use algorithms and data structures
- Write robust, well-documented, and unit-tested programming code
- Experience working on a large-scale software project
- Be confident sharing and collaborating on code over GitHub

As part of the 'R-World' extended programming project, students will have the opportunity to work collaboratively with other classmates. Students are also welcome to come up with another programming exercise to carry out; this is an excellent opportunity to explore something that interests you in greater detail, or for graduate students to get help with that "last chapter"...

2 Course materials

No text-books are required for this course, but a laptop computer (Windows, Mac, or Linux) with RStudio version $\geq$ 0.99.903, `Python` version $\geq$ 3.5.2, and GitHub Desktop installed is required. Advice and help getting these tools installed will be provided ahead of the start of class. All lecture handouts and practical instructions will be downloaded using GitHub Desktop.

3 Attendance & participation

Attendance of all lectures is compulsory; we are covering a lot of ground in a compressed period of time and students who miss one session may not be able to catch back up. Students who miss more than one session may be dropped from the course, unless they can give a good reason (with evidence if necessary) for missing that session.

4 Assessment

All assessment will be via coursework. One week after each section (see 'schedule' for the sections), you will be required to submit your completed programming exercises to the instructor for evaluation. For many of these exercises, you will know how you will have scored before you submit them: they will be graded using the unit testing procedures you will be taught in the first two weeks of the course. The relative contributions of each section are shown below:

Section	Percentage
R	25%
R-World	25%
Python	25%
Options	10%
GitHub	10 %
Participation	5%

Undergraduates taking the 4XXX series class will be required to complete fewer exercises from each section than graduate students completing the 6XXX series class.

The "Honor System" of Utah State applies to this coursework. I strongly advise you *not* to copy-paste answers to exercises from the Internet or from the work of other students. Not only are solutions from outside sources (and other students) often wrong, they are also quite simple for me to detect and those who cheat will be punished in accordance with Utah State regulations.

5 Schedule

This class will consist of the same number of contact hours as a regular class, but compressed into the second half of the semester. As such, each week contains two three-hour sessions.

The course is split into four main sections. In the first (weeks 1–2), you will learn the main concepts of programming in the `R` language, and in the second (weeks 3–4) you will have the opportunity to put these into practice in an extended programming exercise. In this extended exercise ('R-World'), you will be guided through the steps of writing an `R` package that contains a simulation of an ecosystem, complete with terrain-generation, dispersing plants, and animal herbivores. In the third section (weeks 5–6), we will focus on algorithms and data-structures, which are the foundations of computer science. These sections will be completed in the programming language Python; changing language may seem daunting, but you will find that the parallels between `R` and `Python` (and, indeed, many programming languages) are so numerous that the transition will be swift. Finally, in the last section (week 7) we will be covering some optional topics whose content *you* will help determine by feedback with the instructor during the course. While I have two activities prepared, I am happy to change these according to the wishes of the class.

Week	Title	1st session	2nd session
1	**Basic** `R`	calculation	scripting
2	**Intermediate** `R`	classes	functional coding
3	**R-World 1**	terrain	plants
4	**R-World 2**	animals	packaging
5	**Algorithms 1**	basic `Python`	sort algorithms
6	**Algorithms 2**	data structures	optimization algorithms
7	**Option week**	`C++` overview	`Shiny` apps

6 Miscellanea

- I will be teaching this class remotely. This means that all lectures and practicals will be conducted over Skype, and all office hours meetings will also be conducted over Skype.
- ADA compliance: Students with physical, sensory, emotional or medical impairments may be eligible for reasonable accommodations in accordance with the Americans with Disabilities Act and Section 504 of the Rehabilitation Act of 1973. All accommodations are coordinated through the Disability Resource Center in Room 101 of the University Inn, 797-2444 voice, 797-0740 TTY, or toll free at 1-800-259-2966. Please contact

the DRC as early in the semester as possible. Alternate format materials (Braille, large print or digital) are available with advance notice.

- Sexual harassment is defined by the Affirmative Action/Equal Employment Opportunity Commission as any unwelcome sexual advances, requests for sexual favors, and other verbal or physical conduct of a sexual nature. If you feel you are a victim of sexual harassment, you may talk to or file a complaint with the Affirmative Action/Equal Employment Opportunity Office located in Old Main, Room 161, or call the AA/EEO Office at 797-1266.
- Students whose religious activities conflict with the class schedule should contact me at the beginning of the semester to make alternative arrangements.
- Course scheduling and structure may change at short notice with no warning.
- After the introductory session on `Python`, students will be given the option of continuing to complete the course in `R`. I strongly advise students to try to learn `Python`; I may be more lenient in marking the work of those who have attempted difficult exercises in `Python`, but will not penalize the work of anyone who continues working in `R`. Course materials will be geared towards those using `Python`, of course, but students will find it trivial to adapt them to `R`.