E-207: Systems Development for Computational Science

Course Overview and Contact Information

Course Overview
Semester and Year: Fall 2020
Day/Time: On demand
Format: Online
Course Website: https://harvard-iacs.github.io/2020-CS107/ (check often for latest updates)

Instructor Contact Information
Name: David Sondak
Office/Office hours: Maxwell Dworkin G111 / TBD
Phone: 617-384-6994
Email: dsondak@seas.harvard.edu

Course Description and Learning Objectives
Computation has emerged as the third pillar of science alongside the pillars of theory and experiment. Computational science is maturing rapidly and has found considerable and significant use in supporting scientists from various disciplines (including all engineering disciplines, physics, chemistry, finance, biology, and data analysis to name a few). Many burgeoning scientists are still taught to write "a code" for some problem and to debug when things look wrong. Given the ever-increasing complexity of software solutions to scientific problems, this old paradigm is no longer tenable.

E-207 is an applications course highlighting the use of software engineering and computer science in solving scientific problems. You will learn the fundamentals of developing scientific software systems including abstract thinking, the handling of data, and assessment of computational approaches: all in the context of good software engineering practices.

Course Materials

Course Materials
All updated material requirements can be found on the course website: https://harvard-iacs.github.io/2020-CS107/pages/resources.html

No textbook is required for this course.

You should have access to a computer for exercises and all assignments.
Grading

Please refer to the course website for specifics regarding grading: https://harvard-iacs.github.io/2020-CS107/pages/syllabus.html

0% Skills check
20% Class Participation – Pair-programming exercises, Piazza discussions, class discussions
30% Assignments – You are expected to write high-quality, readable, and tested code. You should strive to do things the right way and think about aspects such as reusability and error handling. You are also expected to document your code.
50% Final Project

Expectations and Policies

Attendance

Optional Lectures: You are encouraged, but not required, to attend the weekly lecture sessions. You are required to watch the recorded lecture before attending a pair-programming session.

Mandatory Sections: You must attend at least one pair-programming session per week. These programming sessions will be hosted by members of the teaching staff. Please note that you are also required to either attend or watch the weekly lecture before attending a pair-programming session. The pair-programming section schedule will be released at the start of the semester.

Accessibility

The Extension School is committed to providing an accessible academic community. The Accessibility Office offers a variety of accommodations and services to students with documented disabilities. Please visit https://www.extension.harvard.edu/resources-policies/resources/disability-services-accessibility for more information.

Academic Integrity/Honesty

You are responsible for understanding Harvard Extension School policies on academic integrity (https://www.extension.harvard.edu/resources-policies/student-conduct/academic-integrity) and how to use sources responsibly. Not knowing the rules, misunderstanding the rules, running out of time, submitting the wrong draft, or being overwhelmed with multiple demands are not acceptable excuses. There are no excuses for failure to uphold academic integrity. To support your learning about academic citation rules, please visit the Harvard Extension School Tips to Avoid Plagiarism (https://www.extension.harvard.edu/resources-policies/resources/tips-avoid-plagiarism), where you'll find links to the Harvard Guide to Using Sources and two free online 15-minute tutorials to test your knowledge of academic citation policy. The tutorials are anonymous open-learning tools.
The course topics include:

Unix / Bash scripting
Python language features
Automatic differentiation (final project topic)
Tools (packaging, documentation, containers)
Design patterns, datastructures, and algorithms
Databases

There will also be an optional C++ component.

For exact details, policies, and the most up-to-date schedule, please consult the course website: https://harvard-iacs.github.io/2020-CS107/