CS 510 - Computing for Scientists
Fall 2015
Course Syllabus

Instructor: Dr. Justin Dressel (dressel@chapman.edu)

Time and Place: Tuesday 7:00 - 9:50pm, Hashinger Science Center 222

Office Hours: After class and by appointment

Overview: CS 510 is a graduate-level course intended to introduce modern computing tools and techniques to science-oriented students from diverse backgrounds. Assuming little prior knowledge, students will become proficient with a powerful set of inter-operable tools that are suitable for problem-oriented and data-intensive applications now common in modern science. While emphasizing the trichotomy of data (structuring, processing, and visualization), students will explore efficient algorithm design, industry-best software development practices, and the implementation of numerical techniques. Students will be expected to complete programming assignments in freely available languages such as Python, Julia, R, Javascript, and C.

Units: CS 510 is a 3 unit course

Prerequisites: Graduate standing or permission of instructor

Required Text: N/A

Course materials: All course materials will be made available via the course site on Blackboard when possible. Blackboard will be used for submitting assignments, viewing grades, etc.

Homework, Exams, and Grading: Homework will consist of literate programming assignments, which should be neatly typeset as professional notebook presentations. All supplemental material should be made available via a linked git repository. Absolutely no late work will be accepted. The midterm and final will be more involved programming projects that build on the preceding homework assignments. Homework will count for 30% of the course grade, the midterm project for 30%, and the final project for 40%. To receive a passing grade in the course you must receive a passing grade (>60%) in every component (assignments and projects) of the course.

Collaboration Policy: I encourage you to discuss and study course material together. However, all work you submit for this course must be your own. Non-original material must be properly cited in a README file turned in with your assignments. Any incidents of academic misconduct will be dealt with severely in accordance with the Chapman University Academic Integrity policy (see below).

Overall Graduate Program Objectives:

1. Graduates will develop quantitative reasoning skills, which will enable them to: (a) formulate and solve problems using extrapolation, approximation, and rational estimation, while maintaining precision, accuracy, and statistical validity; and, (b) create and refine quantitative models describing natural phenomena.
2. Graduates will develop critical thinking, end-to-end problem solving, and data-analysis skills by applying the principles of computational science to scientific problems. With these skills, they will be able to: (a) collect, process, and analyze data, (b) use applied mathematics and computational tools to both pose and solve scientific problems.
3. Graduates will leverage advanced computing technology to solve scientific problems, including: (a) manipulating and analyzing data on high performance computer systems (e.g., clusters and supercomputers), and (b) constructing solutions to scientific programs that take advantage of massively parallel algorithms and efficient data structures.

Specific Course Objectives:

1. Competently manipulate files, directories and utilities in the Unix operating system.
2. Be able to set up a programming environment, showing basic familiarity with:
    a. Console editors (e.g., Emacs, Vim)
    b. Modern editors (e.g., Atom, Sublime-text)
    c. Interactive interpreters and notebooks (e.g., Jupyter, Beaker)
    d. Compilation frameworks (e.g., cmake, scons)
    e. Version control (e.g., git, hg)
    f. Debugging and profiling tools (e.g., gdb, gprof)
3. Demonstrate basic familiarity with data formats (e.g., CSV, YAML, JSON, HDF)
4. Explore available programming languages for data processing and simulation, including:
    a. Natively compiled languages (e.g., C, Go)
    b. Just-in-time compiled languages (e.g., Julia, Javascript)
    c. Interpreted languages with compiled libraries (e.g., Python, R)
5. Demonstrate familiarity with data visualization libraries (e.g., matplotlib, dygraphs, vega, D3)
6. Understand the foundations of machine architecture, machine representations of data, efficient algorithms, data structures, and dominant programming paradigms.
7. Demonstrate the basic principles of software engineering, such as test-driven development, modular design, extensibility, verification, validation, optimization, and refactoring.
8. Understand basic simulation techniques, such as numerical methods for linear algebra, differential equations, cellular automata, and Monte Carlo simulations.
9. Understand the concept of high performance and embarrassingly parallel computation.

Diversity and Equality: Chapman University is committed to ensuring equality and valuing diversity. Students and professors are reminded to show respect at all times as outlined in Chapman's Harassment and Discrimination Policy: http://tinyurl.com/CUHarassment-Discrimination. Any violations of this policy should be discussed with the professor, the Dean of Students and/or otherwise reported in accordance with this policy.

Chapman University's Academic Integrity Policy: "Chapman University is a community of scholars that emphasizes the mutual responsibility of all members to seek knowledge honestly and in good faith. Students are responsible for doing their own work and academic dishonesty of any kind will be subject to sanction by the instructor/administrator and referral to the university Academic Integrity Committee, which may impose additional sanctions including expulsion. Please see the full description of Chapman University's policy on Academic Integrity at http://www.chapman.edu/academics/academicintegrity/index.aspx."

Chapman University's Students with Disabilities Policy: "In compliance with ADA guidelines, students who have any condition, either permanent or temporary, that might affect their ability to perform in this class are encouraged to contact the Disability Services Office. If you will need to utilize your approved accommodations in this class, please follow the proper notification procedure for informing your professor(s). This notification process must occur more than a week before any accommodation can be utilized. Please contact Disability Services at (714) 516–4520 or visit http://www.chapman.edu/students/student-health-services/disability-services if you have questions regarding this procedure or for information or to make an appointment to discuss and/or request potential accommodations based on documentation of your disability. Once formal approval of your need for an accommodation has been granted, you are encouraged to talk with your professor(s) about your accommodation options. The granting of any accommodation will not be retroactive and cannot jeopardize the academic standards or integrity of the course."