CS-0254: Genetic Programming

COURSE INFORMATION

Instructor Info:

Lee Spector

lasCCS@hampshire.edu

Office Extension: x5352

Office Hours: Regular office hours: Tuesdays 10:00–11:30, Wednesdays 1:00–2:30, and Thursdays 2:00–3:30. Other times can be set up by arrangement (in person or via email). Sign up for regular office hours, advising day meetings, and occasionally other signup times on Moodle here.

Term: 2014S

Meeting Info:

Tuesday	12:30 PM – 01:50 PM Adele Simmons Hall (ASH) 126
Thursday	12:30 PM – 01:50 PM Adele Simmons Hall (ASH) 126

Description:

Genetic programming is a computational technique that harnesses the mechanisms of natural evolution

-- including genetic recombination, mutation, and natural selection -- to synthesize computer programs automatically from input/output specifications. It has been applied to a wide range of problems spanning several areas of science, engineering, and the arts. In this course students will explore several variations of the genetic programming technique and apply them to problems of their choosing. Prerequisite: One programming course (any language)

Course Objectives:

- To understand and apply genetic programming, a computational problem-solving technique based on evolutionary principles.
- To conduct and present the results of independent project work.
- To develop computer programming fluency

- To develop computer programming fluency.

Evaluation Criteria: Students will be evaluated on the basis of class participation, including weekly "show and tell" presentation assignments, a portfolio that includes code for a large-scale project, and a project presentation. It is essential that all students attend every class session except in cases of illness or genuine emergencies. It is also expected that students will, through their programming work and their class participation, demonstrate both an understanding of the material covered in class and growth with respect to programming skills. Students falling significantly short of these expectations -- for example, students with more

than one unexcused absence -- should not expect to receive evaluations.

Additional Info:

Texts and Other Materials

The required texts are:

- Poli, R., W. B. Langdon, and N. F. McPhee. 2008. *A Field Guide to Genetic Programming*. ISBN 978-1-4092-0073-4. The book is freely downloadable under a Creative Commons license as a PDF from http://www.gp-field-guide.org.uk/ and low cost printed copies can be purchased from lulu.com. Because the PDF is freely available the Hampshire bookstore will not be ordering hardcopies, but they would be happy to special order you a copy if you contact them at bookstore@hampshire.edu.
- Halloway, S., and A. Bedra. 2012. *Programming Clojure, Second Edition.* ISBN 978-1-934356-86-9. Pragmatic Bookshelf.

We will be programming in the Clojure programming language, using the Counterclockwise integrated development environment and in some cases the leiningen build tool. Many other environments are available for developing Clojure code and students are welcome to use whatever they wish, but only Counterclockwise usage will be supported in class.

We will learn about Clojure and genetic programming from the texts listed above and also

from the professor's:

- Book chapters on "Genetic and Evolutionary

Computation," "Genetic Programming," and "Evolution of Complex Programs" from *Automatic Quantum Computer Programming: A Genetic Programming Approach*
- Conference paper on "The Push3 execution stack and the evolution of control"
- clojinc: examples and problem sets
- evolvesum: A simple binary genetic algorithm in Clojure, to demonstrate one way to write an evolutionary loop
- gp: demonstration Clojure code for genetic programming
- clojush: implementation of the PushGP genetic programming system.

We will also use a variety of Clojure-related online resources. Students should become familiar with all of these sites at the beginning of the course, and use them as needed:

- Introduction to Clojure – Modern dialect of Lisp (Part 1)
- tryclj
- 4clojure
- clojuredocs
- clojureatlas

Facilities

Students may use their own computers, the Macs in ASH 126, and the high performance computing cluster in the Hampshire College Cluster Computing Facility. Students should not expect files left on the Macs in ASH 126 to persist; the discs on those machines may be erased without notice at any time. Students may find it convenient to use a thumb drive or cloud storage to transport files to and from class.

Division I Distribution Credit

Successful completion of this course satisfies the Divison I distribution requirement in Mind, Brain, and Information. This course provides opportunities for satisfaction of Division I cumulative skills requirements in Quantitative Skills and Independent Work.

Difficulty/Level

This course is intended to serve students with a wide range of backgrounds. All students are expected to start the course alread knowing how to

program in some language and to be capable of learning a new language relatively quickly. Beyond this, students with little previous experience should resist being intimidated and bear in mind that I take background into account in writing evaluations. Students with extensive previous experience should note that the class is structured to provide ample opportunities for more advanced work.

Work Time Expectations

In this course, students are expected to spend six

to eight hours a week of preparation and work outside of class time. This time includes reading the texts and online sources, programming, developing project ideas, preparing for class presentations, and documenting project work.

Class Format and Schedule

This course will not follow a rigid, pre-specified schedule but will instead be paced to the progress and interests of the students.

The general sequence of activities in the course will be:

1. Learn a shared programming language, Clojure, which will probably be new to most students but which supports the work of this course (and many other kinds of work) particularly well.
2. Learn about the genetic algorithms and genetic programming, including variations of genetic programming that use the Push programming language (one version of which, clojush, is implemented in Clojure).
3. Develop and execute ideas for original genetic programming projects.

These activities are not entirely distinct, and they will be interleaved to some extent throughout the semester.

Class time will be spent on a mixture of:

1. Lecture/demonstration
2. Show and tell sessions
3. Demonic coding sessions

Every Thursday class will begin with a show and tell session, in which each student is expected to briefly present and run recent programming work to the

class, using the projected computer. Each student must present new work every Thursday. Expectations regarding presentation content and format will be discussed in class.

In a demonic coding session the class is split into two groups and the available time is split into two periods. In the first period one of the groups is coders and the other is demons; in the second period the roles are reversed. Coders sit at workstations and work on their own projects for the entire period. Demons rotate among the coders at announced times, observing and interacting with one coder at a time. Demons may ask questions and/or make suggestions, and coders must dedicate a percentage of their time (approximately 50%) to demonic interactions. Each student must have access to his/her current work files every day -- on a laptop computer, or a thumb drive, or a networked server, etc. -- so that he/she will always be ready to participate as a coder in a demonic coding session.

Code sharing is encouraged, although each student is expected to develop a body of code that is mostly original and to explicitly mark all non-original code. See the Plagiarism Policy section below.

Policies in Regards to Illness, Epidemic, or Pandemic

If you have a fever, please stay home, take good care of yourself, and contact me by email or phone. When you are able to work at home you should be able to participate in classes and to submit work electronically. If your illness makes it impossible for you to meet the course deadlines then contact me and we will negotiate an accommodation.

Plagiarism Policy

Official policy text:

> All Hampshire College students and faculty, whether at Hampshire or at other institutions, are bound by the ethics of academic integrity. The entire description and college policy can be found in Non Satis Non Scire at handbook.hampshire.edu under Academic Policies/Ethics of Scholarship. Plagiarism is the representation of someone else's work as one's own. Both deliberate and inadvertent misrepresentations of another's work as your

own are considered plagiarism and are serious breaches of academic honesty and integrity. All sources used or consulted in the process of writing papers, examinations, preparing oral presentations, course assignments, artistic productions, and so on, must be cited. Sources include material from books, journals or any other printed source, the work of other students, faculty, or staff, information from the Internet, software programs and other electronic material, designs and ideas.

All cases of suspected plagiarism or academic dishonesty will be referred to the Dean of Advising who will review documentation and meet with student and faculty member. Individual faculty, in consultation with the Dean of Advising, will decide the most appropriate consequence in the context of the class. This can range from revising and resubmitting an assignment to failing the course. Beyond the consequence in the course, CASA considers first offenses as opportunities for education and official warning. Multiple or egregious offenses will have more serious consequences. Suspected instances of other breaches of the ethics of academic integrity, such as the falsification of data, will be treated with the same seriousness as plagiarism and will follow the same process.

In this course we will often be sharing and borrowing code. This is an important aspect of the course and an important aspect of modern programming practice. This does not mean, however, that it is acceptable to submit code that is not your own without acknowledging sources. Sources should be clearly and explicitly provided in everything that you produce.

Incompletes

Course incompletes are restricted and governed by College policy, and will be negotiated on an individual basis.