



CS-0284: Evolutionary Computation

Home ► CS-0284-1_2016S

Turn editing on

COURSE INFORMATION



Instructor(s):

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 10:00-11:30. 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.

TA(s):

Ben Bailey

bfb14@hampshire.edu

Office Hours:

Term:

2016S

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:

Evolutionary computation techniques harness the mechanisms of biological evolution, including mutation, recombination, and selection, to build software systems that solve difficult problems or shed light on the nature of evolutionary processes. In this course students will explore several evolutionary computation techniques and apply them to problems of their choosing. The technique of "genetic programming," in which populations of executable programs evolve through natural selection, will be emphasized. Prerequisite detail: One college-level programming course, in any language.

Course Objectives:

- To gain familiarity with the concepts and computational techniques of evolutionary computation.
- To apply several of these concepts and techniques to hands-on research and development activities.
- To develop skills in "functional"-style programming.
- To conduct independent programming-based project work.
- To develop project presentation skills.
- To work collaboratively with classmates.

Evaluation Criteria:

You will be evaluated on the basis of:

- Attendance (consistent, on time, participating)
- Code (reflecting engagement, growth, and understanding of course materials)
- Results (well documented)
- Presentations (3 x 3min + 1 x 6min)
- Portfolios (4, containing code, results, and self-eval maps)

Self-eval maps:

```
{:attendance "This sentence describes my attendance and participation."
:code "This sentence describes the code that I wrote."
:results "This sentence describes results my code produced."
:presentation "This sentence describes what I presented."
:knowledge "This text describes what I learned."}
```

Missed, late, or inadequate portfolios or presentations will be noted in your evaluation. If you fail to submit more than one portfolio or to give more than one presentation, then you should not expect to receive an evaluation.

Additional Info:

Division I Distribution Credit

Successful completion of this course satisfies the Division 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.

Texts

We will read parts of:

- Clojure for the Brave and True: Learn the Ultimate Language and Become a Better Programmer, by Daniel Higginbotham, No Starch Press, 2015, ISBN: 978-1-59327-591-4. Available in hardcopy from the Hampshire bookstore or free online at: <http://www.braveclojure.com>
- A Field Guide to Genetic Programming, by Riccardo Poli, William B Langdon, and Nicholas Freitag McPhee, Lulu.com, 2008, ISBN 978-1-4092-0073-4. Available in hardcopy from the Hampshire bookstore or free online at: <http://www.gp-field-guide.org.uk>

Additional readings/videos are linked in the Materials section.

Software

We will use the Clojure programming language. Two development setups will be supported in the class:

- Leiningen + plain text editor + Gorilla REPL
- Counterclockwise

Links for these tools are in the Materials section, and installation procedures will be discussed in class.

Many other Clojure development environments exist. You can use another one if you really want to, but if you do so you should not expect any support for that usage from the professor or TA. You may **NOT** use an environment without the feature that automatically re-indent Clojure code. Code that is not properly indented should never be included in a portfolio, shown in a presentation, or sent to the professor or TA with a request for help. You must work in an editor with automatic re-indentation and you must use that feature consistently to keep your code indented properly.

Facilities

You may use your own computer and/or the Macs in ASH 126, which will be available at various posted hours. You 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.

Difficulty/Level

This course is intended to serve students with a wide range of backgrounds, including students with only one previous programming course (in any language) and students with significant computer science and programming experience. Students with little previous experience should resist being intimidated by the more difficult readings, etc., and bear in mind that I take background into account in writing evaluations. If you find a reading to be difficult, try to extract the gist of it (which may be all that you need at this point in your learning) and talk to me if you want to understand more. Students with extensive previous experience should note that the class is structured to provide ample opportunities for more advanced work; feel free to talk to me about ideas for projects, etc.

Demonic Coding

Many class sessions will be dedicated in part or entirely to "demonic coding" (<http://faculty.hampshire.edu/lspector/demonic.html>). You must therefore bring your current work files every day -- on a laptop computer, or a thumb drive, or a networked server, etc. -- and always be ready to participate as a coder in a demonic coding session.

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. If your illness makes it impossible for you to meet the course deadlines then contact me and we will negotiate an accommodation.

Adaptations and Accommodations

If you need course adaptations or accommodations because of a disability, or if you have a medical condition that may impact your performance or participation in this course, then please let me know. If you have approved accommodations then please go to Accessibility Services in CASA/Lemelson Center to pick up Letters of Accommodation to facilitate a proactive discussion about reasonable accommodations for this course. If you have documented disabilities but have not already already contacted Accessibility Services are encouraged to so. Accessibility Services can be contacted via email: Accessibility@hampshire.edu, via phone: 413-559-5498, or in person: Lemelson Center (CASA entrance).

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.

Schedule

Portfolio due dates are:

- Tuesday, February 9
- Thursday, March 10
- Tuesday, April 5
- Friday, April 29

Presentation dates are:

- Tuesday, February 9
- Thursday, March 10
- Tuesday, April 5
- Tuesday, April 26 or Thursday, April 28

Aside from these due dates, the course will not follow a rigid, pre-defined schedule. We will move through materials and topics in a way that will be responsive to our collective progress and interests. Readings and viewings will be announced in class.

That said, the flow of topics in the course will be roughly:

- Clojure
- Functional programming
- Variation
- Selection
- Genetic algorithms
- Genetic programming
- Artificial life



News forum



Social forum

Materials



4clojure



And now, digital evolution



The Annual "Humies" Awards



Automatic Quantum Computer Programming: A Genetic Programming Approach
(selected chapters)



Beating the Averages



A Bunch of Clojure Quil examples based on Daniel Shiffman's Nature of Code



The calculator problem and the evolutionary synthesis of arbitrary software



Clojinc: Steps toward Clojure, starting from zero



Clojure Atlas



The ClojureBridge Curriculum



Clojure Cheatsheets



Clojure.org



Clojure for the Brave and True



Clojure for Java Programmers Part 1 - Rich Hickey



Clojure for Java Programmers Part 2 - Rich Hickey



Clojure for Lisp Programmers Part 1 - Rich Hickey



Clojure for Lisp Programmers Part 2 - Rich Hickey



Clojure from the ground up



Clojure Google Group



Clojure Reddit



Clojure Style Guide



Clojure Videos



clojush: implementation of the PushGP genetic programming system



Counterclockwise



Decomplecting Clojure



Division Blocks and the Open-Ended Evolution of Development, Form, and Behavior



Emergence of Collective Behavior in Evolving Populations of Flying Agents



Evolving Virtual Creatures With Genetic Algorithms (Karl Sims)



Evolutionary Computing with Push



Evolution of Artificial Intelligence



evolvesum: A simple binary genetic algorithm in Clojure































Extensible Data Notation



A Field Guide to Genetic Programming



From evolutionary computation to the evolution of things

-  [Genbebob](#)
-  [Genetic Algorithms and Evolutionary Computing \(The Nature of Code\)](#)
-  [General Program Synthesis from Examples Using Genetic Programming](#)
-  [Genetic Programming \(Langdon, McKay, and Spector\)](#)
-  [Genetic Programming and Evolvable Machines](#)
-  [The Genetic Programming Bibliography](#)
-  [Genetic Programming in Clojure](#)
-  [Genetic Stability and Territorial Structure Facilitate the Evolution of Tag-mediated Altruism](#)
-  [Gorilla Incanter](#)
-  [Gorilla REPL](#)
-  [gp: demonstration Clojure code for genetic programming](#)
-  [Guidelines for Demonic Coding Sessions](#)
-  [Introduction to Clojure – Modern dialect of Lisp \(Part 1\)](#)
-  [An Introduction to Genetic Algorithms \(Melanie Mitchell\)](#)
-  [Learning: Genetic Algorithms \(Patrick Winston\)](#)
-  [Leiningen](#)
-  [Planet Clojure](#)
-  [Pucks](#)
-  [The Push3 execution stack and the evolution of control](#)
-  [Push-language Discourse](#)
-  [Quil: Clojure/ClojureScript library for creating interactive drawings and animations](#)
-  [SIGEVolution Tributes to John Henry Holland](#)
-  [Secrets of Gorilla REPL](#)
-  [Selection Songs](#)
-  [Solving Uncompromising Problems with Lexicase Selection](#)
-  [Try Clojure](#)
-  [The Value of Values with Rich Hickey](#)
-  [What is Evolutionary Computation \(David Fogel\)](#)

NAVIGATION




Home


- My home
 - Site pages
 - My profile
 - Current course
 - CS-0284-1_2016S**
 - Participants
 - General
 - Materials
 - My courses

ADMINISTRATION



Course administration

 Turn editing on


 Edit settings


Users

 Filters

Reports

 Grades

 Backup

 Import

Question bank

Switch role to...

My profile settings


PEOPLE





 Participants

QUICKMAIL





 Compose New Email

 Signatures

 View Drafts

 View History

-  Alternate Emails
-  Configuration



LIBRARY INFORMATION



[Library Homepage](#)

ACTIVITIES



-  [Forums](#)
-  [Resources](#)

FILTERED COURSE LIST



Current courses

Fall 2015

Spring 2016

[CS-0284: Evolutionary Computation](#)

Other courses

[All courses ...](#)

 [Moodle Docs for this page](#)

moodle@hampshire.edu

You are logged in as Lee Spector ([Log out](#))

[Home](#)