You are logged in as Lee Spector (Logout)

CS-0254: Genetic Programming

Home 〉 My courses 〉 CS-0254-1_2012F

Turn editing on

COURSE INFORMATION

Instructor Info:	**Lee Spector** lasCCS@hampshire.edu Office Extension: x5352 Office Hours: Regular office hours: Tuesdays 9:30-11:00, 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 Hampedia here.
TA Info:	**Zeke Nierenberg** zan09@hampshire.edu **Alec Goebel** acg10@hampshire.edu Office Extension: x4356
Term:	2012F
Meeting Info:	**Tuesday** 02:00 PM - 03:20 PM Adele Simmons Hall (ASH) 126 **Thursday** 02:00 PM - 03:20 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.
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.
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

NAVIGATION

Home

- My home
  - Site pages
  - My profile
  - My courses
    - CS-0262-1_2013S
    - CS-104T-1_2010F
    - CS-0236-1_2010F
    - CS-0148-1_2010F
    - CS-0311-1_2011S
    - CS-0148-1_2011F
    - CS-0201-1_2011F
    - CS-0254-1_2011F
    - CS-0109-1_2012F
    - **CS-0254-1_2012F**
      - Participants
      - Reports
      - General
    - CS-104T-1_2013F
    - CS-0303-1_2013F

SETTINGS

- Course administration
  - Turn editing on
  - Edit settings
  - Users
  - Filters
  - Grades
  - Backup
  - Restore
  - Import
  - Reset
  - Question bank
  - Legacy course files
- Switch role to...
- My profile settings

PEOPLE

Participants

QUICKMAIL

- Compose New Email
- Signatures
- View Drafts
- View History
- Alternate Emails

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.
- Halloway, S., and A. Bedra. 2012. *Programming Clojure,*

  Second Edition. ISBN 978-1-934356-86-9. Pragmatic Bookshelf.

Additional readings will be distributed in class.

We will be programming in the Clojure programming language, using the clooj lightweight 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 clooj usage will be supported in class.

We will learn Clojure in part by working through the professor's:

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.

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 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

Configuration

MY PRIVATE FILES

No files available

Manage my private files

ACTIVITIES

Resources

LIBRARY INFORMATION

Library Homepage

CURRENT COURSES LIST

Current Courses

CS-0262: Research in Artificial Intelligence : Research in Artificial Intelligence

Future Courses

CS-0303: Unconventional Computing

CS-104T: Cognitive Science Fiction

Other Courses

Cognitive Science Fiction

Creative Programming Workshop

CS-0109: Programming Creativity

CS-0148: Other Minds

CS-0201: Research Experience in Artificial Intelligence

CS-0254: Genetic Programming

CS-0254: Genetic Programming

Other Minds

Research in Artificial Intelligence

This course is intended to serve students with a wide range of backgrounds. 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.

All courses ...

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. Develop fluency in Clojure while also learning about the fundamentals of genetic algorithms and genetic programming.
3. Develop an original genetic programming system in Clojure while also learning about advanced topics in genetic programming, including variations of genetic programming that use the Push programming language (one version of which, clojush, is implemented in Clojure).
4. 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

Hampshire College has a rigorous policy on plagiarism, outlined in detail in the student handbook. As stated in College documents "Plagiarism (from the Latin for 'kidnapper') is a term covering everything from inadvertently passing off as one's own the work of another because of ignorance, time constraints, or careless note-taking, to hiring a ghost writer to produce an examination or course paper." In particular, it covers false citation, false data, intentional poor documentation, papers written by others, unacknowledged multiple authors or collaboration, unacknowledged multiple submission, and other forms of academic dishonesty. The penalties are severe, so you should always be proactive in identifying all sources. When in doubt you should ask your me about what is and isn't appropriate.

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.

Readings from Lee's GP/Quantum Book
Push3 paper at GECCO 2005
Evolving SQL Queries from Examples with Developmental Genetic Programming
Trivial Geography in Genetic Programming
Culture Enhances the Evolvability of Cognition
Ontogenetic Programming
What's in an evolved name? The evolution of modularity via tag-based reference
Tag-based Modularity in Tree-based Genetic Programming

Moodle Docs for this page

You are logged in as Lee Spector (Logout)

Home