CS-0254: Genetic Programming

COURSE INFORMATION

Instructor(s):

Lee Spector
lasCCS@hampshire.edu
Office Extension: x5352
Office Hours:
Regular office hours: Tuesdays 10:30-11:30, Thursdays 10:30-11:30 and 12:30-2:00. I am available at many other times as well; feel free to contact me in person or via email to arrange other meeting times. Sign up for regular office hours and advising day meetings here.

TA(s):

Caitlin Hensley
ch15@hampshire.edu
Office Hours:
Jackson Lipfert
jwl15@hampshire.edu
Office Hours:

Term:

2018S

Meeting Info:

Tuesday	09:00 AM - 10:20 AM Adele Simmons Hall (ASH) 126
Thursday	09:00 AM - 10:20 AM 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 detail: One programming course (in any language).

Course Objectives:

- To understand and apply genetic programming, a computational problem-solving technique based on evolutionary principles.
- To develop skills in "functional"-style programming.
- To conduct independent programming-based project work.
- To work collaboratively with classmates.

Evaluation Criteria:

You will be evaluated on the basis of attendance, participation (both in class and in out-of-class activities), and a portfolio of code and text. More detail on the expectations for participation and portfolios is provided below.

You are expected to demonstrate through your participation, code, and text that you have read and thought about the course readings, that you have developed facility with the code environment used in the class, and that you have engaged with several of the class topics at the implementation level.

You should expect a positive evaluation or a high grade if you engage consistently with the course material, have a strong record of attendance and participation, and produce a complete portfolio that demonstrates growth and understanding. Significant lapses in participation or gaps in portfolios will be noted in evaluations and count against grades.

These criteria will be applied individually to each student, without regard to how other students in the class perform. This course should not be considered a competition. It is understood that students will be entering the course with a wide range of backgrounds, and students will be evaluated in the context of those backgrounds.

If you have concerns about evaluations or grades then you are encouraged to check in with me about your standing as the course proceeds.

Additional Info:

Texts

Clojure for the Brave and True by Daniel Higginbotham. No Starch Press, 2015. ISBN-10: 1593275919, ISBN-13: 978-1593275914. Available free online

A Field Guide to Genetic Programming by Riccardo Poli, William B. Langdon, and Nicholas F. McPhee. Lulu Enterprises, 2008. ISBN-10: 1409200736, ISBN-13: 978-1409200734. Available free online.

All other readings will be available from the class website.

Programming Environment

We will learn and use the Clojure programming language, editing and running code with Gorilla REPL and managing projects with Leiningen. All of this software is free and runs on multiple platforms.

You may use your own computer -- we will help everyone to get the software running on their own computers in class -- and/or the computers in ASH 126, which have the software installed and will be available at various posted hours. Note that the computers in ASH 126 may be erased without notice at any time; keep copies of your files somewhere else!

Many Clojure editing and development environments exist, aside from Gorilla REPL. You can use another one if you really want to (as long as it supports automatic re-indentation of Clojure code, ***which is mandatory and you should use it often in whatever you use***), but if you use another environment we may not be able to provide support for it.

Clojure Resources

- Clojure.org, the main site for the Clojure language; note especially their Clojure resources page
- Clojure cheatsheet
- Clojure style guide
- Clojure TV
- Two blog posts on dealing with Clojure error messages (stack traces): 1 and 2
- Lee's tutorial materials:
  - Clojestions, suggested exercises for learning Clojure
  - Clojinc, a saved REPL session intended to support semi-independent learning of Clojure
- Other tutorial materials:
  - 4Clojure, interactive Clojure problems
  - Learn X in Y minutes, Where X=clojure, a brief introduction to Clojure

Portfolios

Your portfolio should contain:

- A summary: Less than one page, describing the contents of your portfolio and orienting the reader with references to specific items of interest. You should submit new versions of your summary periodically; only the most recently submitted summary will be considered part of your portfolio.
- Hacks: Files containing new code, submitted on at least five days of each week (with exceptions to be announced in class). See below for more information.
- Comments: Files containing new comments, submitted on at least five days of each week (with exceptions to be announced in class). See below for more information.
- RICE reports: Brief descriptions of each RICE session (approximately one per week), saying who you met with, when and where, and what you discussed/did during the session. See below for more information.
- Projects: No specific number of projects is required, and there is no mandatory schedule for submissions.

Schedule

Relatively little class time will be dedicated to lectures. Most class sessions will consist of 20 minutes of critiques (see below), followed by 60 minutes of "demonic coding" (see below).

We will not follow a rigidly defined schedule, but will progress through the following stages over the semester:

- Functional programming in Clojure
- Evolutionary computation
- Genetic programming
- Project work

The work in portfolios is expected to progress through these stages as well.

Hacks

Hacks are pieces of original Clojure code that run without errors, ideally submitted in Gorilla REPL worksheets.

Each hack can be small -- even a single line of code -- but each should do something new relative to those that came before it.

The expectation is that hacks early in the semester will be simple Clojure exercises, while hacks later in the semester will incorporate genetic programming concepts and applications.

Within these general outlines you are free to hack as you wish, but it is suggested that you begin with hacks that respond to my Clojestions, the exercises in my Clojinc tutorial, or the problems at 4Clojure.

Comments

Comments are English text, submitted in plain text files. Each comment can be small -- even a single sentence -- and comments can be about anything related to the course. For example, some comments may reflect thoughts or questions about the readings, while others may describe recent or planned hacks or projects.

RICE: Required Immersive Collaborative Experience

Roughly weekly, you will be directed to engage in a Required Immersive Collaborative Experience (RICE) activity.

Student pairings will be random. If you find that you have a partner who is not sufficiently cooperative in setting up a meeting, let me know at least two days before the deadline so that alternative arrangements can be made in time for you to stay on schedule.

RICE reports in your portfolio should be brief descriptions of who you met with, when and where, and what you discussed or did during the session.

Demonic Coding

Most class sessions will be dedicated in part or entirely to demonic coding. You must therefore have access to 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.

Critiques

In each critique session, two or three students will be chosen at random and asked to show one or two things from their recent hacks, comments, or projects. Others in the class (including the professor and TAs) will make observations and ask questions. This will be informal, and you needn't have prepared presentations for critiques, but you should always have in mind which of your recent pieces of work you might show if called upon.

Policies

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.

ILLNESS

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 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, the I encourage you to do so. Accessibility Services can be contacted via email: Accessibility@hampshire.edu, via phone: 413-559-5498, or in person at CASA.

PLAGIARISM

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.

To Do

Things that you should read/do to prepare for specific class sessions will be listed here.

Portfolio: Summary

Less than one page, describing the contents of your portfolio and orienting the reader with references to specific items of interest. Submit new versions of the summary periodically; only the most recently submitted summary will be considered part of your portfolio.

Portfolio: Hacks

Submit a file containing new code on at least five days of each week (with exceptions to be announced in class).

Portfolio: Comments

Submit a file containing new comments on at least five days of each week (with exceptions to be announced in class).

Portfolio: RICE Reports

Submit a brief description of each RICE session (approximately one per week), saying who you met with, when and where, and what you discussed/did during the session.

Portfolio: Projects

Submit projects here, as they emerge in your work. No specific number of projects is required. When submitting a new version of a previously-submitted project, make it clear in the title that it replaces the earlier version. Projects should be described in your portfolio summary.

Announcements