CS-0109: Programming Creativity

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 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.

Term: 2014F

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: This course is an introduction to computer science and programming framed by the question, "Is it possible for a computer to be creative?" The core areas of computer science will be introduced, including algorithms, complexity, computability, programming languages, data structures, systems, and artificial intelligence, with an eye toward the insights that they can provide about issues of computational creativity. Students will complete several programming projects to demonstrate developing technical skills and engagement with the themes of the course. No previous experience with computers or with programming is required.

Course Objectives:
- To understand the core concepts and areas of study in computer science.
- To learn how to program, or how to program better, and more generally to develop the ability to turn abstract ideas into running code.

- To appreciate the significance of computational models of cognition and creativity.
- To develop and present a software portfolio.
- To work collaboratively with classmates.

Evaluation Criteria: Each student is expected to attend consistently, to read all assigned readings, to participate actively in class discussions, and to complete the following assignments:

First Code	A short programming assignment to demonstrate basic understanding of the *Processing* programming environment. *Details* 1. Do **one or both** of the following: ◦ Produce and submit a Processing program that creates a different but not completely random pattern of colored shapes each time it is run. ◦ Produce and submit a Processing program that uses my turtle graphics code but draws something else interesting, preferably variable. 2. Be prepared to demonstrate and discuss your work in class on the due date. Procedures for submitting work will be covered in class.
First Code Presentation	In-class presentation of the first code assignment; approximately 3 minutes.
Processing Tool	*Processing* code that may help in the production of creative programs by others in the class. *Details* 1. Produce and submit a Processing program that you think may be useful in the construction of a larger program that demonstrates computational creativity. 2. Be prepared to demonstrate and discuss your work in class on the due date.
Tool Presentation	In-class presentation of the tool project; approximately 3 minutes.

Final Project	A significant *Processing* program that exhibits machine creativity. *Details* 1. Produce and submit a Processing program that exhibits machine creativity, preferably using one or more tools developed earlier in the course (by yourself or by others). 2. Be prepared to demonstrate and discuss your work in class on the presentation dates.
Project Presentation	In-class presentation of the final project; approximately 5 minutes.

Students are urged to consult with the professor before starting the Tool or the Final Project. Course evaluations will briefly describe the completed projects; students should bear this in mind when choosing the topics and forms for their projects.

Additional Info:

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.

Texts

- *The Pattern on the Stone: The Simple Ideas that Make Computers Work*, by Daniel Hillis, 1998.
- *Getting Started with Processing*, by Casey Reas & Ben Fry. 2010.

Software

The *Processing* programming environment, available from http://processing.org/

Lee's Processing examples (zipped): http://hampshire.edu/lspector/courses/lee_processing_examples.zip

Facilities

Students may use their own computers and/or the Macs in ASH 126, which will be available at various posted hours. The *Processing* programming environment can be run under Mac OS X, Windows, or Linux, and students can work on whatever platform they prefer, but all submitted work must run under Mac OS X. 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.

Difficulty/Level

This course is intended to serve students with a wide range of backgrounds, including students with no programming experience 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 a reading or a class discussion is over your head, 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." 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. -- 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. 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.

Schedule

The following is subject to change. Adjustments will be announced in class.

Tuesday 12:30-1:50 PM	Thursday 12:30-1:50 PM
	September 4 First class **In class:** Introductions, syllabus, overview
September 9 **Before class:** • **Read:** Hillis preface and the "Talking to the Computer" section of chapter 3; R&F chapters 1 & 2 • **Explore:** processing.org **In class:** • Discuss First Code assignment • Introduction to Processing (dots, turtle, etc.) • Demonic coding	**September 11** **Before class:** • **Read:** Lem, Trurl's Electronic Bard; R&F chapter 3 **In class:** • Demonic coding
September 16 **In class:**	**September 18** ***ADVISING DAY*** (no class)

• First Code assignment presentations **Due:** • First Code assignment	
September 23 **Before class:** • **Read:** Hillis chapters 1 & 2 **In class:** • Electrons to algorithms • Discuss Tool Project • Demonic coding	**September 25** **In class:** • Demonic coding
September 30 **Before class:** • **Read:** R&F chapters 4 & 5 **In class:** • Text, numbers, control • Demonic coding	**October 2** **Before class:** • **Read:** Boden Precis **In class:** • Creativity and computational art • Demonic coding
October 7 **Before class:** • **Read:** R&F chapters 6 & 7 **In class:** • Notations and control • Demonic coding	**October 9** **In class:** • Tool & final project brainstorming • Demonic coding
October 14 ***OCTOBER BREAK*** (no class)	**Before class:** • **Read:** R&F chapters 8, 9, & 10 **In class:** • Data structures • Functions

	• Demonic coding
October 21 **In class:** • Demonic coding	**October 23** **In class:** • Demonic coding
October 28 **In class:** • Demonic coding	**October 30** **In class:** • Tool assignment presentations **Due:** • Tool assignment
November 4 **Before class:** • **Read:** Hillis unread sections of chapter 3 + chapter 4 **In class:** • Computatbility • Demonic coding	**November 6** **In class:** • Quantum computing • Demonic coding
November 11 **Before class:** • **Read:** Hillis chapter 5 **In class:** • Complexity • Demonic coding	**November 13** **In class:** • Final project discussion • Demonic coding
November 18 **Before class:** • **Read:** Hillis chapter 6 **In class:** • Number systems • Demonic coding	**November 20** **Before class:** • **Read:** Hillis chapter 8 **In class:** • Artificial Intelligence • Demonic coding
November 25	**November 27**

Before class: • **Read:** Hillis chapter 9 **In class:** • Evolutionary computing • Demonic coding	***THANKSGIVING*** (no class)
December 2 **Before class:** • **Read:** McCaffrey and Spector **In class:** • AI & creativity • Demonic coding	**December 4** **In class:** • Demonic coding
December 9 **In class:** • Final project presentations	**December 11** **In class:** • Final project presentations **Due:** • Final project • Portfolio • Self evaluation