CS-0109: PROGRAMMING CREATIVITY

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:

Alec Goebel  
acg10@hampshire.edu  
Office Extension: x4356

Term: 2012F

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.

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.

Everything should be turned in to a folder on the "gibson" file server with your name as the folder name. Each project should include an exported applet folder so that you can demonstrate your applet by directing a web browser to [http://gibson.hampshire.edu/~cs109/](http://gibson.hampshire.edu/~cs109/) and navigating to the proper folder. Procedures for exporting applets and uploading them to gibson will be covered in class, and are documented [here](#).

<table>
<thead>
<tr>
<th>First Code Presentation</th>
<th>In-class presentation of the first code assignment; approximately 3 minutes.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processing Tool</strong></td>
<td>Processing code that may help in the production of creative programs by others in the class.</td>
</tr>
</tbody>
</table>
| **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.  
Everything should be turned in to a folder on gibson with your name as the folder name. |

<table>
<thead>
<tr>
<th>Tool Presentation</th>
<th>In-class presentation of the tool project; approximately 3 minutes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final Project</td>
<td>A significant <em>Processing</em> program that exhibits machine creativity.</td>
</tr>
</tbody>
</table>
| **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 due date. |

https://moodle.hampshire.edu/course/view.php?id=2444
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 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**


**Software**


Lee's Processing examples (zipped):
[http://hampshire.edu/lspector/courses/lee_processing_examples.zip](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 can use the class folder on the "urza" file server to store files; access instructions will be provided in class. 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 computer experience and students with significant computer 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 need 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. -- so that he/she will 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**

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 notetaking, 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.

**Schedule**

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

<table>
<thead>
<tr>
<th>Tuesday 12:30-1:50 PM</th>
<th>Thursday 12:30-1:50 PM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>September 6</td>
</tr>
<tr>
<td>First class</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Before class</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------------</td>
</tr>
</tbody>
</table>
| September 11 |  - Read: Hillis preface and the "Talking to the Computer" section of chapter 3; R&F chapters 1 & 2  
              - Explore: processing.org  
              - Discuss First Code assignment  
              - Introduction to Processing (dots, turtle, etc.)  
              - Demonic coding  |  
| September 13 |  - Read: Lem, Trurl's Electronic Bard; R&F chapter 3  
              - In class:  
              - Demonic coding  |  
| September 18 |  - First Code assignment presentations  |  
| September 20 |  - Due:  
              - First Code assignment  |  
| September 25 |  - Read: R&F chapters 4 & 5  |  
| September 27 |  - In class:  
              - Text, numbers, control  
              - Demonic coding  |  
| October 2   |  - Read: R&F chapters 6 & 7  |  
| October 4   |  - In class:  
              - Notations and control  
              - Demonic coding  |  
| October 9   |  - OCTOBER BREAK (no class)  
              - Before class:  
              - Read: R&F chapters 8, 9, & 10  |  
| October 11  |  - In class:  
              - Data structures  |
Functions
Demonic coding

October 16
In class:
- Demonic coding

October 18
In class:
- Demonic coding

October 23
In class:
- Demonic coding

October 25
In class:
- Tool assignment presentations
Due:
- Tool assignment

October 30
Before class:
- Read: Hillis unread sections of chapter 3 + chapter 4

November 1
ADVISING DAY (no class)

November 6
In class:
- Computability
- Demonic coding

Before class:
- Quantum computing
- Demonic coding

November 8
Before class:
- Read: Hillis chapter 5

November 13
In class:
- Complexity
- Demonic coding

Before class:
- Final project discussion
- Demonic coding

November 15
Before class:
- Read: Hillis chapter 6

November 20
Before class:
- Read: Hillis chapter 8

November 22
THANKSGIVING (no class)

November 27
Before class:

November 29
Before class:
<table>
<thead>
<tr>
<th>Date</th>
<th>Read: Hillis chapter 9</th>
<th>Read: McCaffrey and Spector</th>
<th>In class:</th>
<th>In class:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Evolutionary computing</td>
<td>AI &amp; creativity</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Demonic coding</td>
<td>Demonic coding</td>
</tr>
<tr>
<td>December 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>December 6</td>
<td></td>
<td></td>
<td></td>
<td>Final project presentations</td>
</tr>
<tr>
<td>December 11</td>
<td></td>
<td></td>
<td>Final project presentations</td>
<td></td>
</tr>
</tbody>
</table>

Due:
- Final project
- Portfolio
- Self evaluation