COSC-211
Data Structures

Section 1

Lee Spector

Registration

- Section 1
- Section 2 (Prof Rager): Same core content, different details and policies
- If not registered but want to be, add on system and email me if you have trouble
- **You must sign sign-in sheet to remain registered**

- Introductions
- Course information
- Data structures

- **Introductions**
- Course information
- Data structures

You

- Name
- Pronouns
- Year
- College if not Amherst
- Major(s) or possible major(s)
- Specific and/or non-computer-science interest(s)

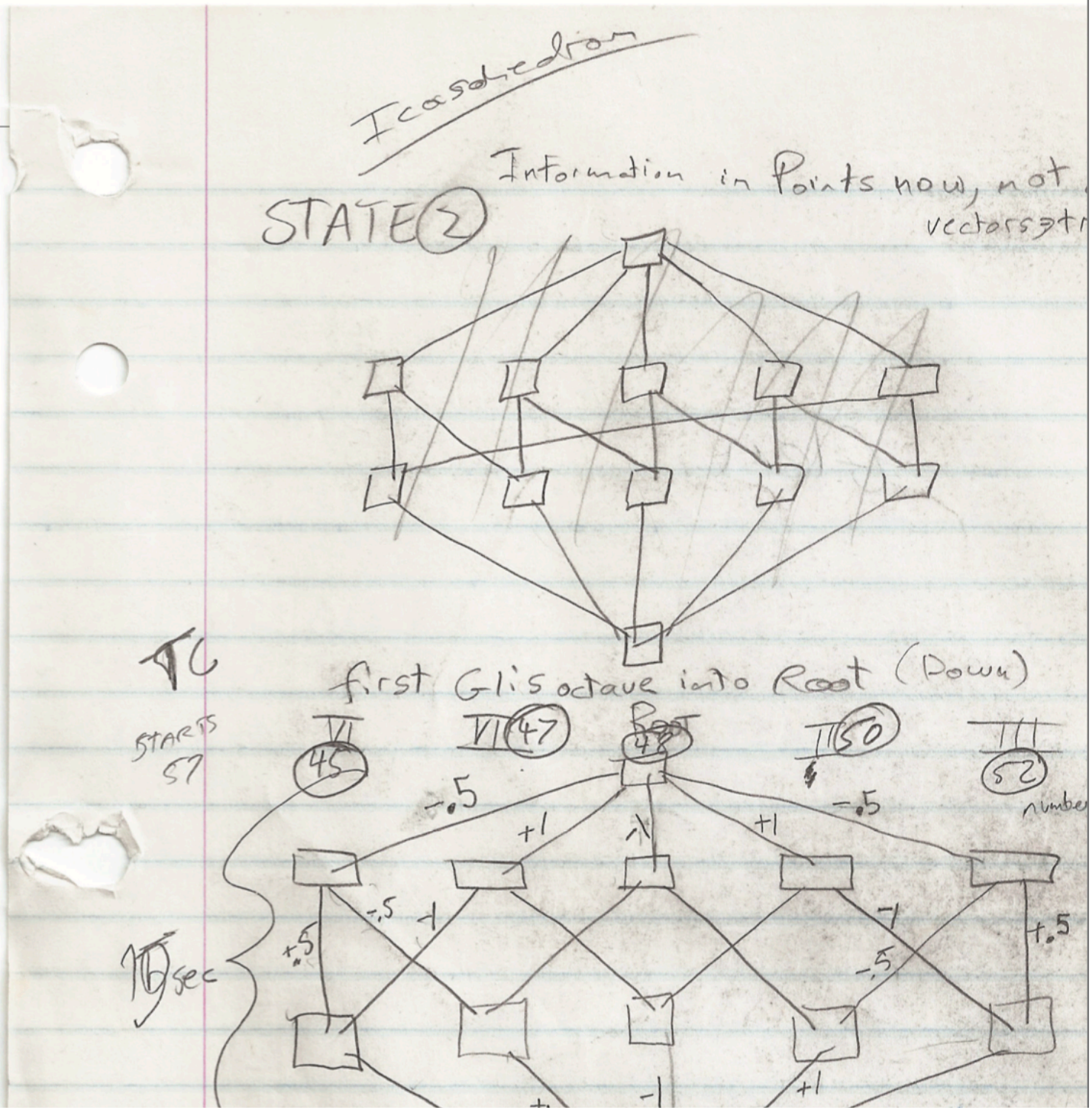
MUSIC

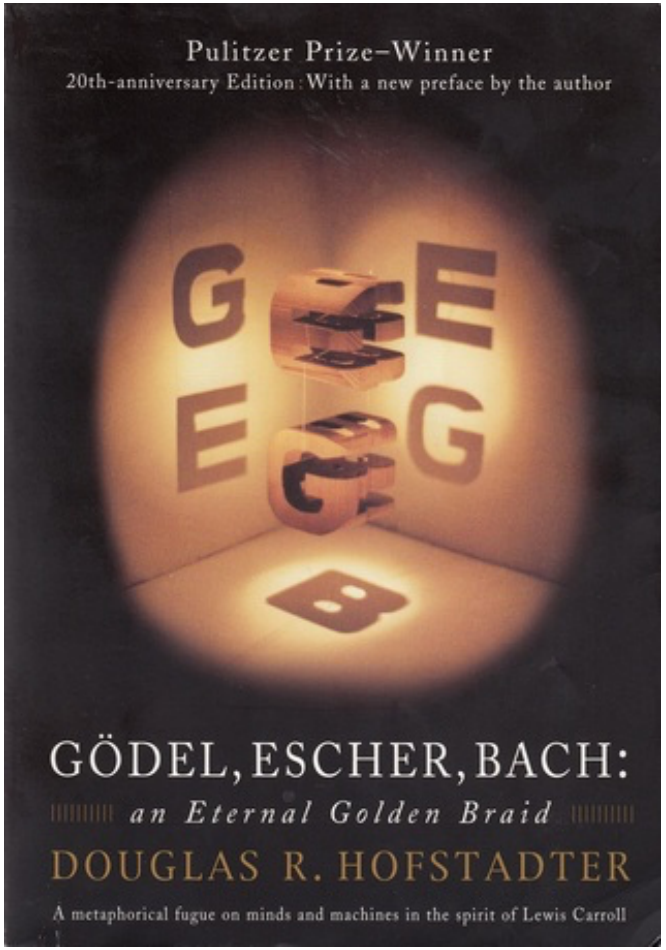
The Spector Sound

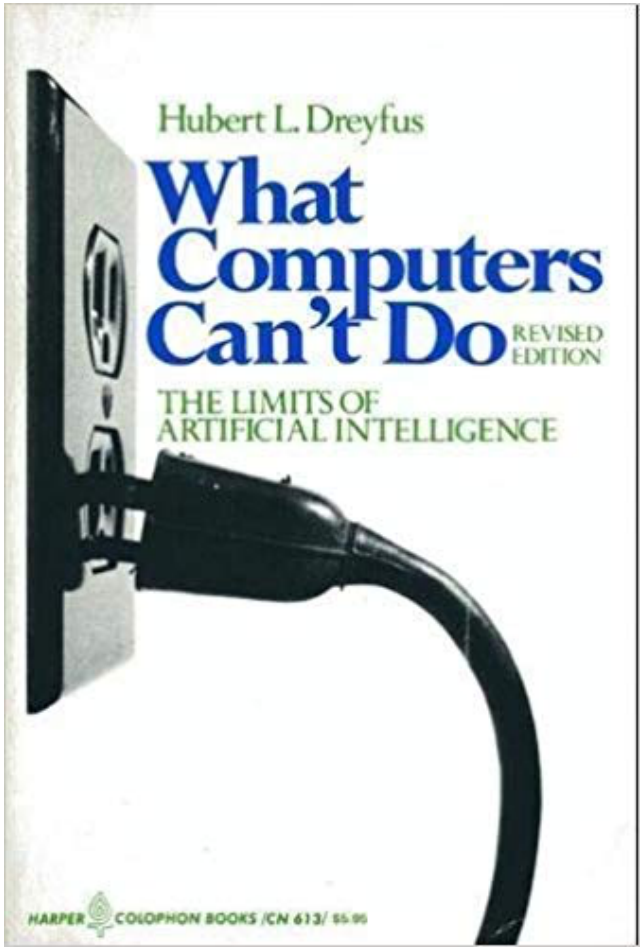
This diagram is a composition called "Geometric Transformations (Nude Brunch)," created in 1981 by Lee Spector '84 in a TIMARA class taught by Gary Lee Nelson. In a windowless room on the fourth floor of Mudd, students in the class composed music by creating code on computer terminals that were connected to a mainframe computer (Spector, a professor of computer science at Hampshire College, believes it was a Xerox Sigma 9) in the building's basement.

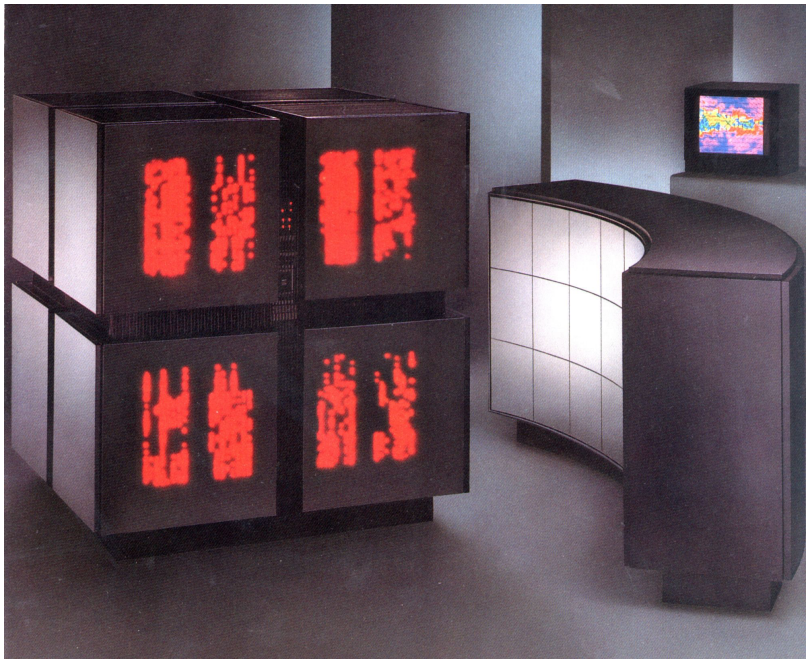
"'Composition' was development of the ideas and the code," he says, "and 'performance' was getting the computer to produce the sound.

"I happened to be a big Buckminster Fuller fan at the time," Spector explains, "and I decided to make a piece that translated the geometric objects he described in his




Pulitzer Prize–Winner
20th-anniversary Edition: With a new preface by the author
GÖDEL, ESCHER, BACH:
an Eternal Golden Braid
DOUGLAS R. HOFSTADTER
A metaphorical fugue on minds and machines in the spirit of Lewis Carroll


Hubert L. Dreyfus
What Computers Can't Do
REVISED EDITION
THE LIMITS OF ARTIFICIAL INTELLIGENCE
HARPER COLOPHON BOOKS /CN 613/ $5.95



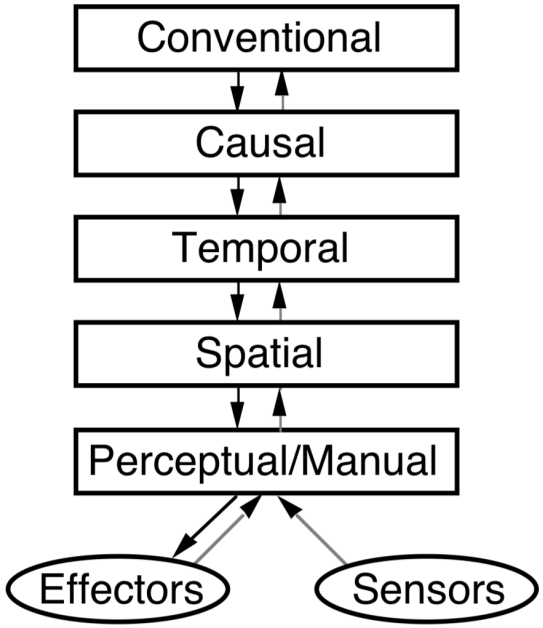


Figure 7. The specific levels of APE.

ELSEVIER

COGNITIVE SCIENCE

Cognitive Science 25 (2001) 941–975

http://www.elsevier.com/locate/cogsci

Partial and total-order planning: evidence from normal and prefrontally damaged populations

Mary Jo Rattermann[a,*], Lee Spector[b], Jordan Grafman[c], Harvey Levin[d], Harriet Harward[e]

[a]*Department of Psychology, Franklin & Marshall College, Lancaster, PA 17604, USA*
[b]*School of Cognitive Science, Hampshire College, Amherst, MA 01002, USA*
[c]*National Institute of Neurological Disorders and Stroke, Building 10; Room 5C205; 10 Center Drive; MSC 1440, Bethesda, MD 20892-1440, USA*
[d]*Department of Physical Medicine and Rehabilitation, Baylor University College of Medicine, 1333 Moursound Avenue, A 205, Houston, TX 77030, USA*
[e]*Callier Center for Communication Disorders, University of Texas at Dallas, 1966 Inwood Road, Dallas TX 75235, USA*

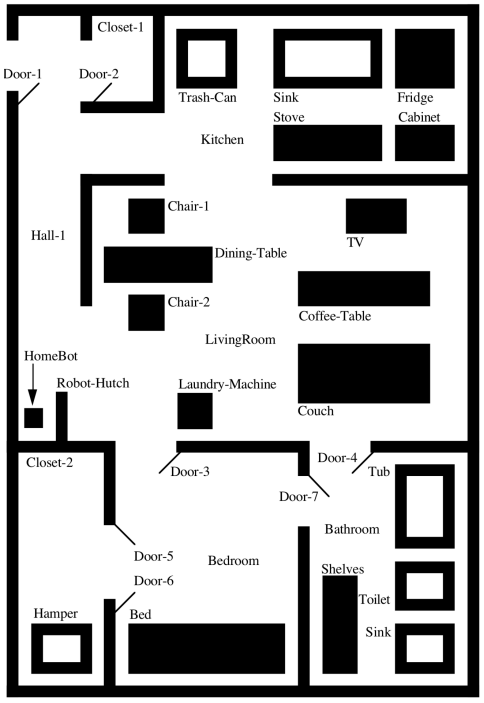
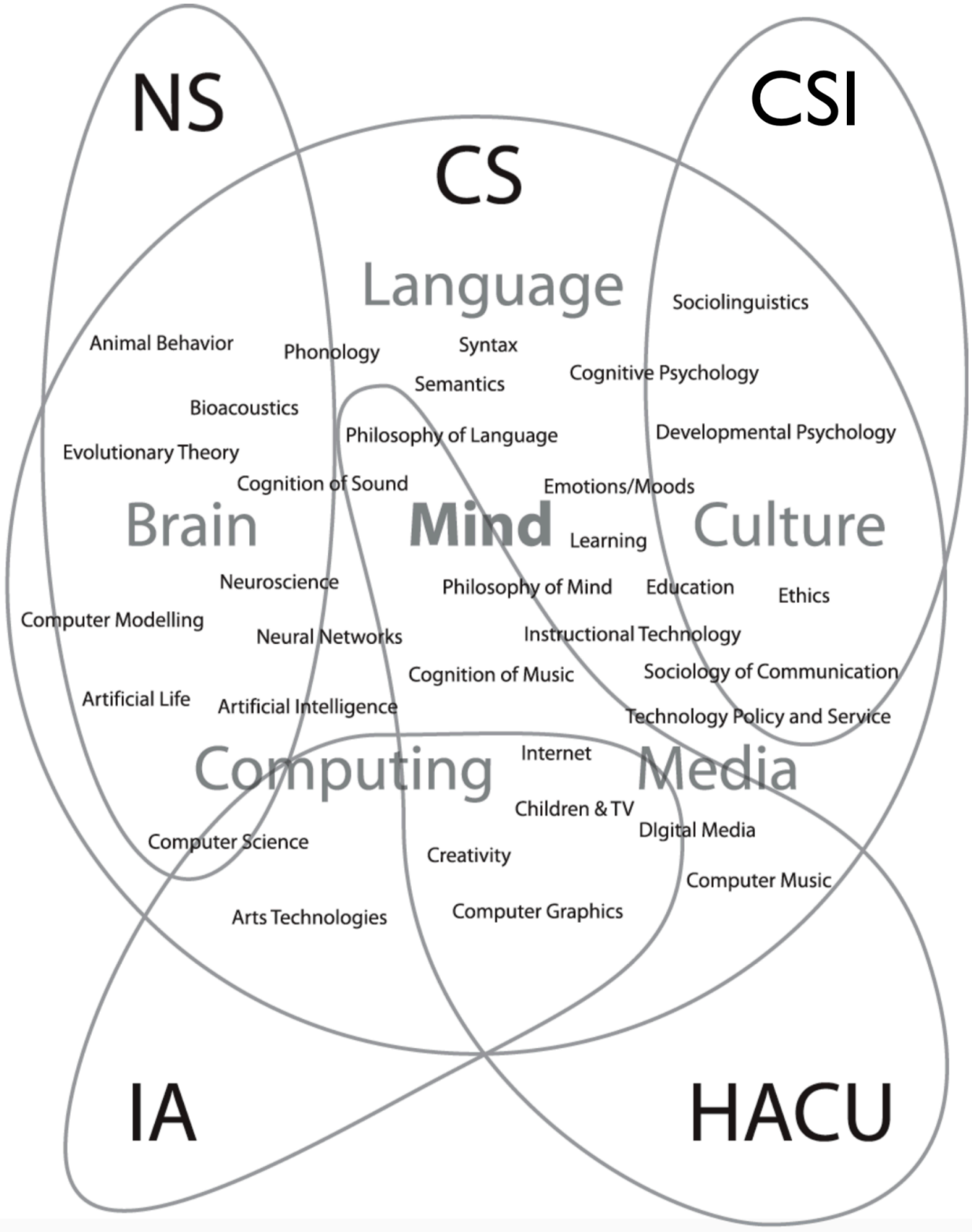


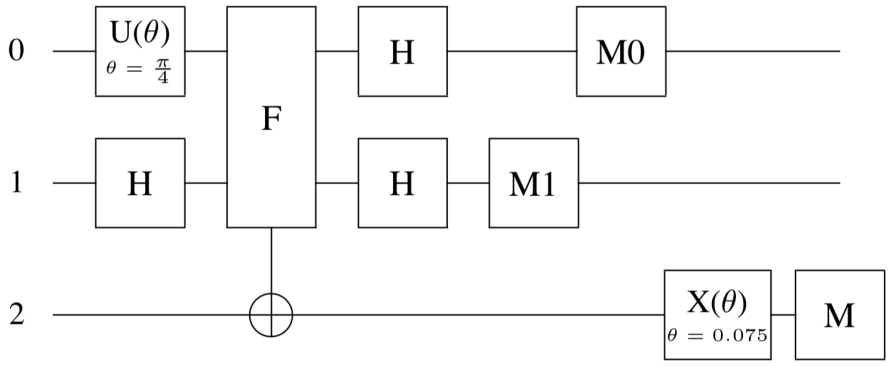
Figure 19. HomeBot's domain.

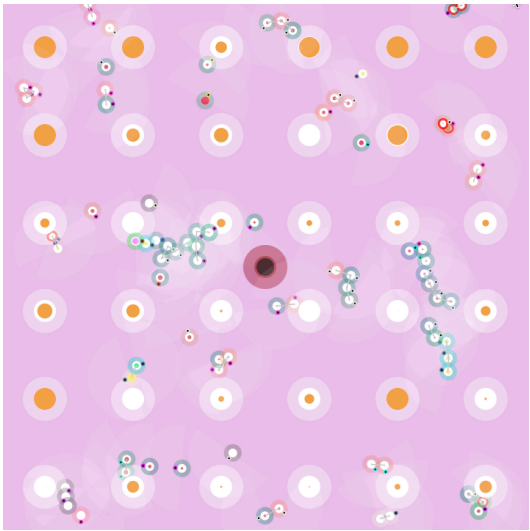

NS
CS
CSI
Language
Sociolinguistics
Animal Behavior
Phonology
Syntax
Cognitive Psychology
Semantics
Bioacoustics
Philosophy of Language
Developmental Psychology
Evolutionary Theory
Cognition of Sound
Emotions/Moods
Brain
Mind
Culture
Learning
Neuroscience
Philosophy of Mind
Education
Ethics
Computer Modelling
Neural Networks
Instructional Technology
Cognition of Music
Sociology of Communication
Artificial Life
Artificial Intelligence
Technology Policy and Service
Computing
Internet
Media
Children & TV
Dlgital Media
Computer Science
Creativity
Computer Music
Arts Technologies
Computer Graphics
IA
HACU

Advanced Topics in Artificial Intelligence
Algorithmic Arts
Animals and Animats: Natural and Artificial Intelligence and Behavior Artificial Intelligence
Artificial Intelligence in 3D Virtual Worlds
Beginning Coding for Science
Biocomputational Developmental Ecology
Code Immersion
Cognitive Science Fiction
Computational Models of Biological Systems
Computer Science Projects
Computing Concepts: Creative Machines?
Creative Programming Workshop
Current Issues in Cognitive Science
Evolutionary Computation
Genetic Programming
Hypertext
Introduction to Artificial Intelligence
Introduction to Cognitive Science
Introduction to Computer Science
Programming Creativity
Programming for Science
Programming Game Theory
Programming Language Paradigms
Quantum Computing with No Prerequisites of Any Kind
Radical Innovation in Digital Arts
Reasoning About Action
Research in Artificial Intelligence
Unconventional Computing
What Computers Can’t Do (limits of computing)
When Machines Talk (natural language processing)

Integrated Teaching & Research

- Undergrad/grad/faculty collaboration
- Wide range of project areas
- Five College research group focusing on evolutionary computing





- Introductions
- **Course information**
- Data structures

Computer Science 211-01: Data Structures

NAVIGATION

Dashboard
Site home
Site pages
My courses
1920S
Computer Science
Data Structures 01 (COSC-211-01-1920S)
Participants
Grades
General
January 26 - February 1
February 2 - February 8
February 9 - February 15
February 16 - February 22
February 23 - February 29
March 1 - March 7
March 8 - March 14

Syllabus

General course information and policies. Be sure to read all of this carefully.

Schedule

This is where you'll find detailed information about what we're doing each day, what you should read and turn in, etc. It will be populated and adjusted incrementally as the course proceeds, so check back frequently.

Announcements

Anonymous Forum

January 26 - February 1

Assignment 1

February 2 - February 8

February 9 - February 15

Syllabus

Amherst College, Spring 2020

COSC-211-01: Data Structures

Instructor: Professor Lee Spector (he/him), SCCE C211, lspector@amherst.edu

Class Meetings: Tuesdays and Thursdays, 10:00-11:20, in SCCE A131

Office hours: Sign up for regular slots (M 1:30-3:30 & 4:30-5:30, Tu 1:30-2:15, Th 12:30-2:15) here. Other times can be arranged by email.

Teaching assistants: Nicholas Carolan and Isaac Caruso, in SCCS A131 Tuesday evenings, 7:00-9:00.

Description: A fundamental problem in computer science is that of organizing data so that it can be used effectively. This course introduces basic data structures and their applications. Major themes are the importance of abstraction in program design and the separation of specification and implementation. Program correctness and algorithm complexity are also considered. Data structures for lists, stacks, queues, dictionaries, sets, and graphs are discussed. This course will provide advanced programming experience.

Texts:

- Open Data Structures (in Java), by Pat Morin. Available online here (click on the links to the Java edition)
- Additional readings that will be distributed on the class Moodle site

Software: IntelliJ IDEA is recommended, but you can use an alternative Java development environment if you wish.

Hardware: IntelliJ IDEA is installed on Science Center computers, but it would be best also to install it on a computer of your own if you are able to do so.

Grading: Midterm exams: 15% and 20% (there will be two and you will get 20% for whichever you do better on), final exam: 25%, homework: 40%. Homework will be due roughly bi-weekly, and will consist of both collaborative assignments (for which you are encouraged to work together) and individual assignments (for which you are forbidden to work together). The homework for the last few weeks of the semester focus on the development of a final project, so the final project will count as several weeks of homework. Partial credit will be awarded for incomplete submissions, but **late submissions will receive no credit at all** (unless required by official accommodations or requested by your class dean). So it is a good idea to plan to submit everything at least a day or two before the deadline, and you should always submit *something* on time even if it is not complete.

Attendance and participation: You are expected attend and participate in all classes, except when you really can't or shouldn't, for example because of illness. While attendance will not be recorded and will not contribute *directly* to your grade, it will nonetheless contribute significantly, *indirectly*, by helping you to master the material, to complete the exercises, and to do well on the exams.

Demonic Coding: In some class sessions we will engage in group work on collaborative assignments. This may include Demonic Coding sessions, which work like this:

- The class is split into "coders" and "demons," and each demon is paired with a coder.
- Coders begin coding on whatever they are working on for the course.
- Demons observe, ask questions, and make suggestions. Demons with fewer skills than their coders can ask more questions, while those with more skills can make more suggestions. Roughly 50% of a coder's time should be devoted to demonic interactions, with the rest devoted to making progress on the code.
- From time to time, demons rotate to other coders.
- Halfway through the session, all coders become demons and all demons become coders.

Demonic coding sessions may be started sometimes without warning, so you should always have access to your code.

Moodle forum: You are strongly encouraged to submit questions and answers to the class Moodle forum. Anonymity is the default, and you are welcome to remain anonymous or to identify yourself depending on the situation. I will not look at identities of anonymous posters unless it is necessary to deal with a problem.

Support: My goal is for you to succeed in this course, to learn a lot and to earn a good grade. If you are having trouble, then I want to know about it and I want to help. Please make proactive use of the support systems built into the course (the Moodle forum, the TAs, and my office hours), and contact me if you may need additional support.

Accommodations: If you have a documented disability that requires accommodations, you will need to register with Accessibility Services for coordination of your academic accommodations. You can reach them via email at accessibility@amherst.edu, or via phone at 413-542-2337. Once you have your accommodations in place, I will be glad to meet with you privately during my office hours or at another time to discuss the best implementation of your accommodations.

Phones and other devices: Feel free to use whatever devices you want if they help you to engage with the class by taking notes, doing searches, executing code, etc., but you should indeed be 100% engaged in the collective work of the class throughout every class meeting.

Honor code: The Amherst College Honor Code applies to this course, as it does to all other Amherst College courses.

Schedule: The details of the schedule will be filled in as we proceed, and posted on the class Moodle site, which you should consult regularly. The general sequence of topics will be:

- Abstract data types and interfaces
- Array and linked list implementations of stacks and queues
- Asymptotic analysis of runtimes and Big-O notation
- Binary trees
- Heaps and priority queues
- Binary search trees
- Hashing
- Dictionaries
- Red-black trees
- Graphs
- Search
- Projects

Schedule

This schedule will be augmented and adjusted as the course proceeds.

Class 1 (Tuesday, January 28)

Before:

- Nothing

In class:

- Introductions
- What we will do in this course and why

Class 2 (Thursday, January 30)

Before:

- Install IntelliJ IDEA
- Read the *Open Data Structures* textbook (ODS) Chapter 1 through 1.6
- Read Assignment 1 (in Moodle section for this week)

In class:

- Abstract data types (interfaces)
- List, queue, stack, and set interfaces
- Big-O notation (briefly)
- Worst-case, amortized, and expected runtimes (briefly)

Class 3 (Tuesday, February 4)

Before:

- Read ODS Chapter 2 through 2.4, and Chapter 3 through 3.2

In class:

- Array implementation of stacks, queues, and deques
- Linked list implementation of stacks, queues, and deques
- Demonic coding

Class 4 (Thursday, February 6)

Before:

- Submit Assignment 1

In class:

- Big-O notation
- Asymptotic analysis

- Introductions
- Course information
- **Data structures**

For some application of interest:

- What data must you store?
- How would you store it?
- How would you add/delete/find/change it?
- How much time and space does that take?
- How does this scale with the amount of data?

Questions?

Reminders

- Install IntelliJ IDEA
- Read the *Open Data Structures* textbook (ODS) Chapter 1 through 1.6
- Read Assignment 1 (in Moodle section for this week)