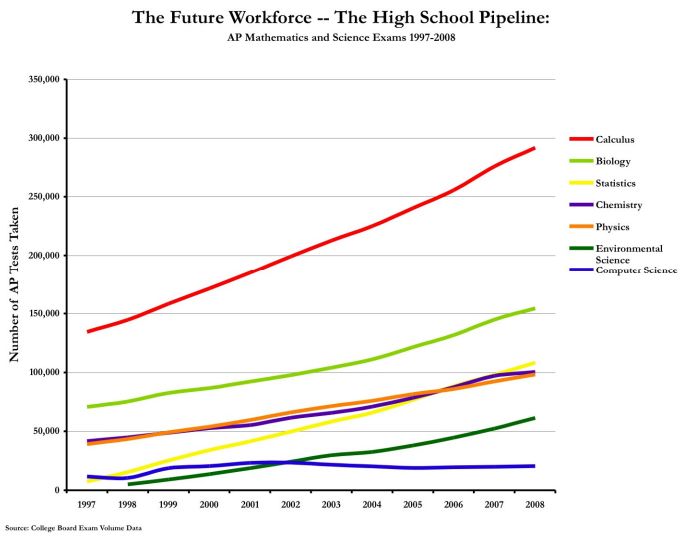
Rethinking the Introduction to Computing

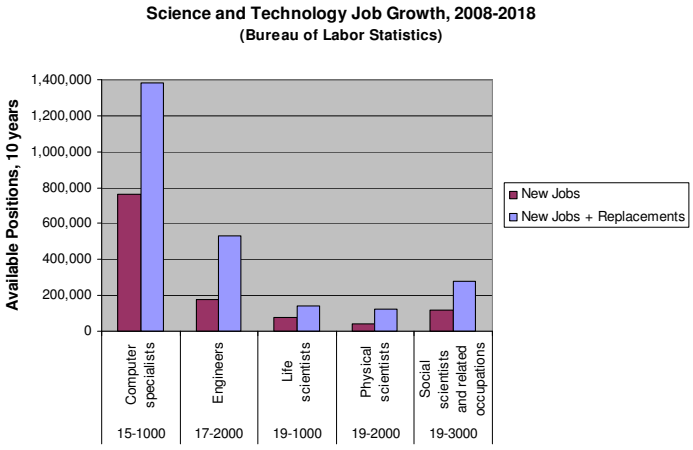
Dr. Tom Cortina
Carnegie Mellon University

The CS Crisis

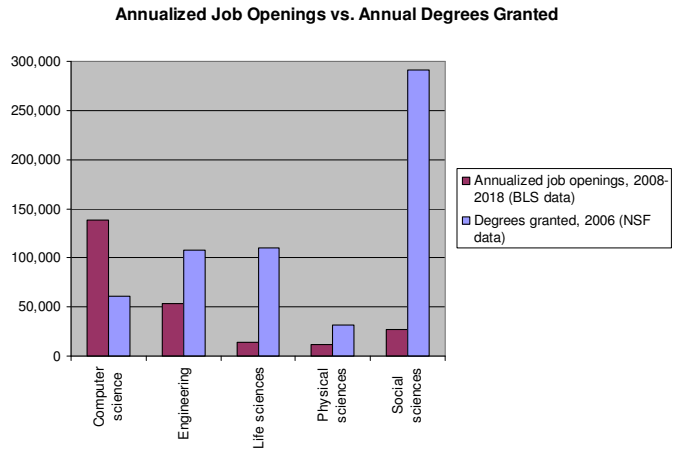
The CS Crisis



The CS Crisis



The CS Crisis



Computational Thinking

- Jeannette Wing publishes a viewpoint article in Communications of the ACM (March 2006)
- Says that computational thinking "represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use."

Computational Thinking

"Computer science is having a revolutionary impact on scientific research and discovery. Simply put, it is nearly impossible to do scholarly research in any scientific or engineering discipline without an ability to think computationally. The impact of computing extends far beyond science, however, affecting all aspects of our lives. To flourish in today's world, everyone needs computational thinking."

"Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent."

Computational Thinking

Computational thinking (CT) deals with using principles of computation to solve complex problems.

CT draws on concepts fundamental to computer science.

- Computer Science is the study of computation (not the study or use of computers).
- Computer Science ≠ Computer Programming (although programming is a part of CS)

What's in a name?

"Computer Science is no more about computers than astronomy is about telescopes."

E.W. Dijkstra (1930-2002)
Dutch computer scientist
Turing Award winner

Examples of CT

- CT is thinking recursively.
- CT is reformulating a seemingly difficult problem into one which we know how to solve.
- CT is choosing an appropriate representation or modeling the relevant aspects of a problem to make it tractable.
- CT is using decomposition in tackling a large complex task.
- CT is prevention, detection, and recovery from worst-case scenarios through redundancy, damage containment, and error correction.
- CT is determining how difficult a problem is to solve.

Jeannette Wing (2008)

CT in daily life

- Looking up a name in a phone book
- Standing in line at a bank or supermarket
- Putting things in your child's knapsack for the day
- Running errands around town
- Cooking a gourmet meal
- Cleaning out your garage
- Storing away your child's Lego pieces scattered on the floor
- Doing laundry or getting food at a buffet
- Doing long division, factoring, GCD, …
- Using sets, drawing family trees, …

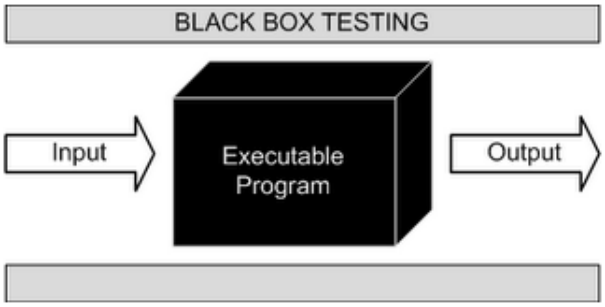
Jeannette Wing (2008)

CT: 2 Big Principles

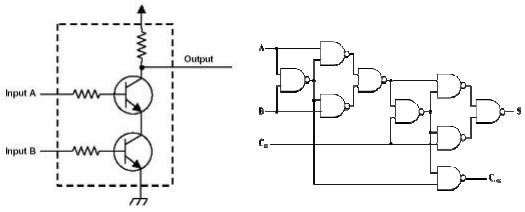
- Abstraction
  - Choosing the right abstractions
  - Operating simultaneously at multiple layers of abstraction
  - Defining the relationships between the layers
- Automation
  - Mechanizing our abstractions, abstraction layers, and their relationships
  - There is some "computer" below (human or machine, virtual or physical)

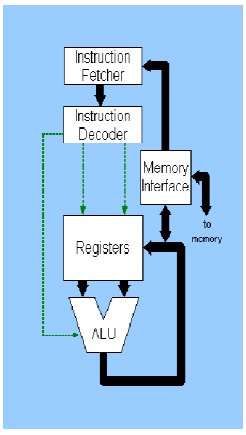
Jeannette Wing (2008)

Abstraction

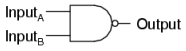


Abstraction

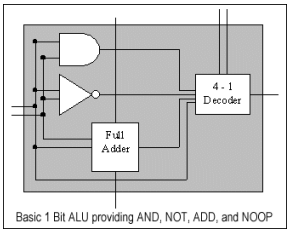




NAND gate

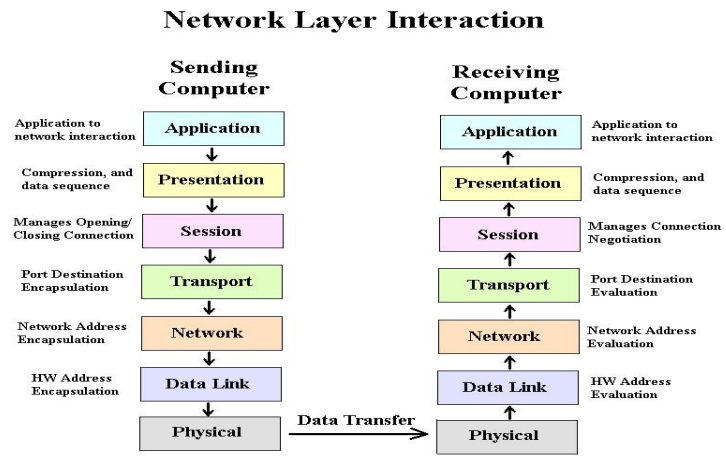


A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

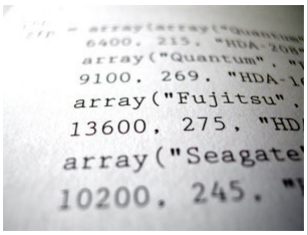


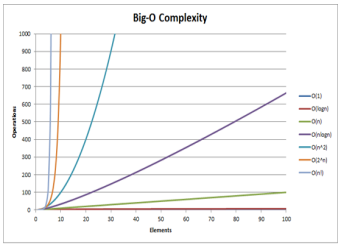
Basic 1 Bit ALU providing AND, NOT, ADD, and NOOP

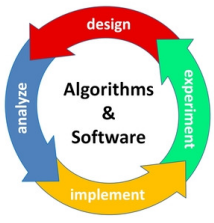
Abstraction



Automation







Machine Learning/Statistics

IBM Jeopardy Challenge

Transportation

Self driving car

Human Computation

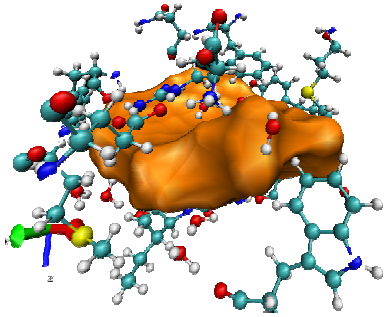
Foiling automated agents

Biology

shotgun sequencing
-mapping the human genome

Chemistry

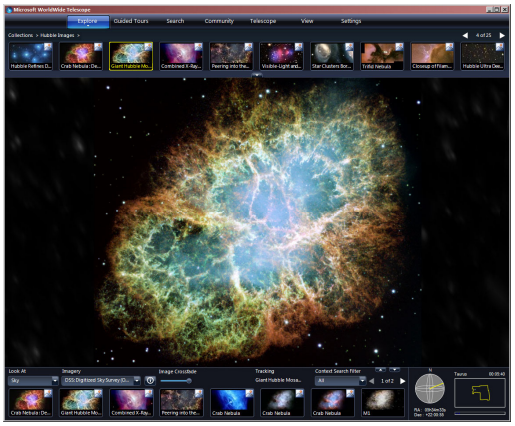
- Optimization and searching algorithms identify best chemicals for improving reaction conditions to improve yields



Computational Chemistry Group, Cleveland State University, OH

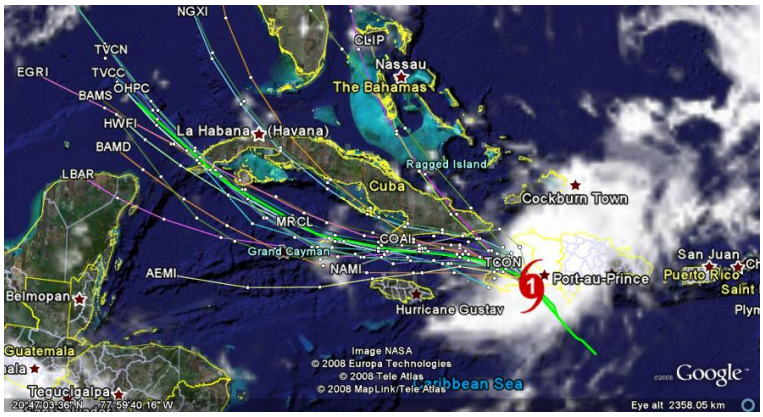
Astronomy

Microsoft WorldWide Telescope



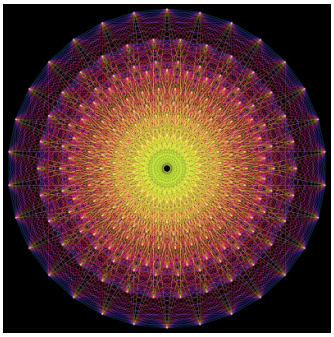
Earth Science

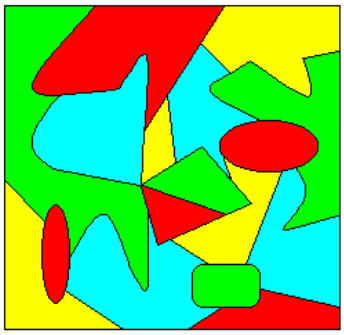
Hurricane Path Prediction



Mathematics/Physics

- Discovering E8 Lie group
- Four-color theorem





Engineering

Boeing 777

Boeing's first fly-by-wire airliner, and its first entirely computer-designed commercial aircraft.

Economics & Finance

- Computerized Stock Trading
- E-commerce (Ad placement, recommendations)

Computational Fields

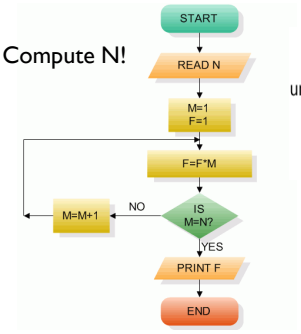
Operational Definition of CT for K-12 Education

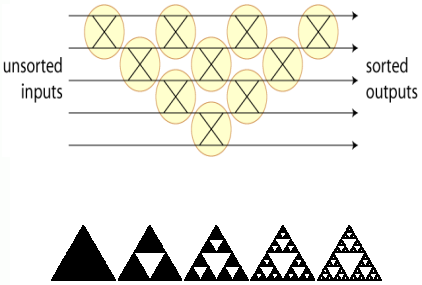
The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA)

CT is a problem-solving process that includes (but is not limited to) the following characteristics:

Operational Definition of CT for K-12 Education

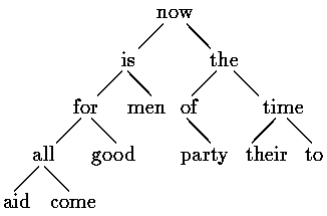
- Formulating problems in a way that enables us to use a computer and other tools to help solve them.

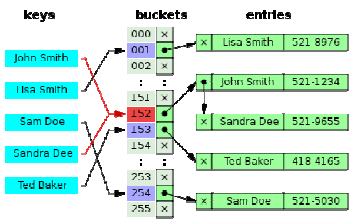


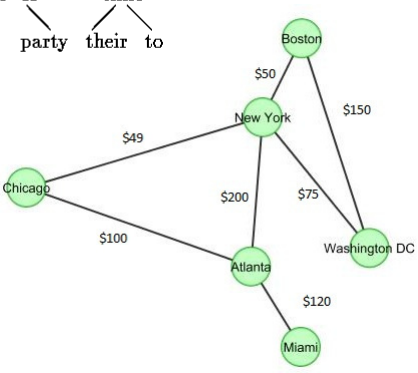


Operational Definition of CT for K-12 Education

- Logically organizing and analyzing data.

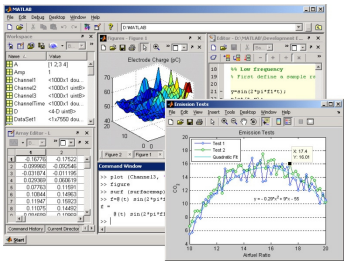


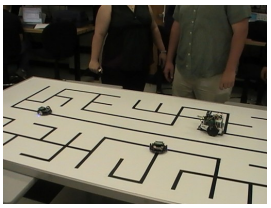




Operational Definition of CT for K-12 Education

- Automating solutions through algorithmic thinking.

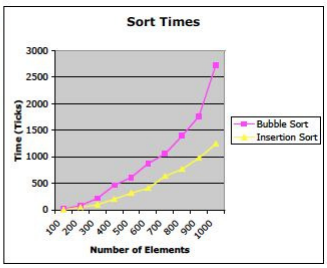




Operational Definition of CT for K-12 Education

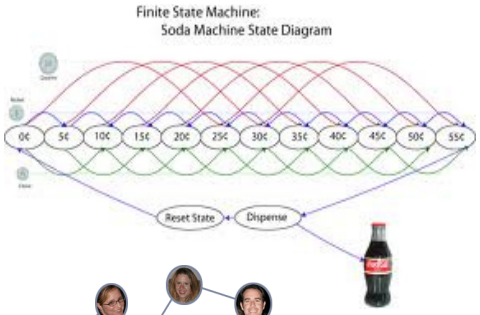
- Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources.

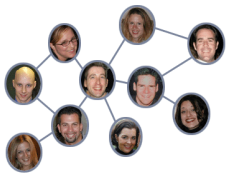
Number of Ele	Bubble Sort	Insertion Sort
100	24	11
200	89	54
300	223	104
400	469	208
500	617	323
600	880	413
700	1060	641
800	1398	771
900	1766	981
1000	2732	1248



Operational Definition of CT for K-12 Education

- Generalizing and transferring this problem solving process to a wide variety of problems.





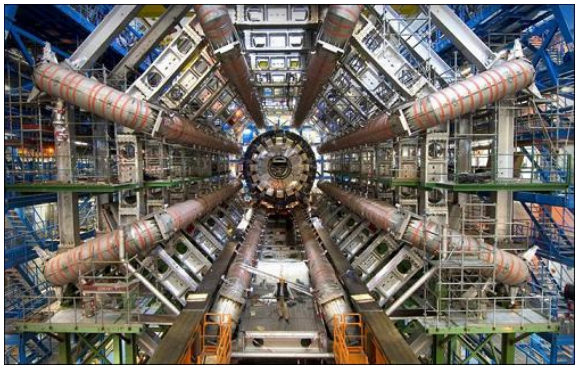
Operational Definition of CT for K-12 Education

These skills are supported and enhanced by a number of dispositions or attitudes that are essential dimensions of CT.

These dispositions or attitudes include:

CT Dispositions and Attitudes

- Confidence in dealing with complexity.

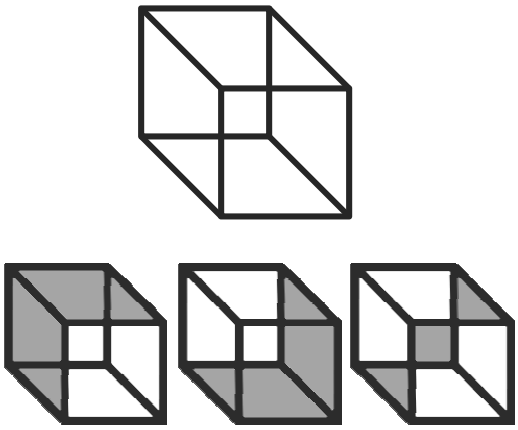


Operational Definition of CT for K-12 Education

- Persistence in working with difficult problems.

Operational Definition of CT for K-12 Education

- Tolerance for ambiguity.



Operational Definition of CT for K-12 Education

- The ability to deal with open-ended problems.

Operational Definition of CT for K-12 Education

- The ability to communicate and work with others to achieve a common goal or solution.

AP CS Principles

- A proposed AP Computer Science: Principles course is intended to foster a wider appeal for the computer science discipline and to better prepare a pipeline of STEM majors.
- Promotes computational thinking.
- Designed around 7 "big ideas".
- Goal: Start new AP course nationally around 2015.

AP CS Principles: Big Ideas

1. Creativity:
Computing is a creative activity.

- Computing fosters the creation of artifacts.
- Computing fosters creative expression.
- Programming is a creative process.

AP CS Principles: Big Ideas

2. Abstraction:
Abstraction reduces information and detail to facilitate focus on relevant concepts.

- A combination of abstractions built upon binary sequences can be used to represent all digital data.
- Multiple levels of abstraction are used in computation.
- Models and simulations use abstraction to raise and answer questions.

AP CS Principles: Big Ideas

3. Data:
Data and information facilitate the creation of knowledge.

- People use computer programs to process information to gain insight and knowledge.
- Computing facilitates exploration and the discovery of connections in information.
- Computational manipulation of information requires consideration of representation, storage, security, and transmission.

AP CS Principles: Big Ideas

4. Algorithms:
Algorithms are used to develop and express solutions to computational problems.

- An algorithm is a precise sequence of instructions for a process that can be executed by a computer.
- Algorithms are expressed using languages.
- Algorithms can solve many, but not all, problems.
- Algorithms are evaluated analytically and empirically.

AP CS Principles: Big Ideas

5. Programming:
Programming enables problem solving, human expression, and creation of knowledge.

- Programs are written to execute algorithms.
- Programming is facilitated by appropriate abstractions.
- Programs are developed and used by people.
- Programming uses mathematical and logical concepts.

AP CS Principles: Big Ideas

6. Internet:
The Internet pervades modern computing.

- The Internet is a network of autonomous systems.
- Characteristics of the Internet and the systems built on it influence their use.
- Cybersecurity is an important concern for the Internet and systems built on it.

AP CS Principles: Big Ideas

7. Impact:
Computing has global impacts.

- Computing affects communication, interaction, and cognition.
- Computing enables innovation in nearly every field.
- Computing has both beneficial and harmful effects.
- Computing is situated within economic, social, and cultural contexts.

AP CS Principles: CT Practices

- Analyzing effects of computation
- Creating computational artifacts
- Using abstractions and models
- Analyzing problems and artifacts
- Communicating processes and results
- Working effectively in teams

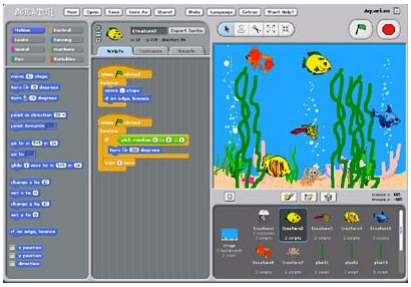
AP CS Principles

- Big Ideas, Key Concepts, and Supporting Concepts
- Learning Objectives and Evidence Statements
- Pilot Sites
  2010 (5 colleges)
  2011 (10 colleges & 10 high schools)
- Attestations: more than 80 colleges and universities support AP CS Principles

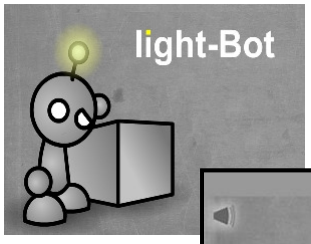
Tools to Introduce CT

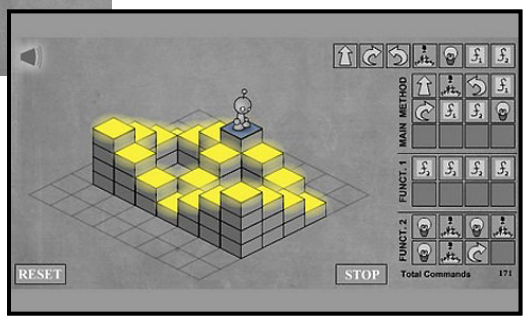
Tools to Introduce CT

SCRATCH



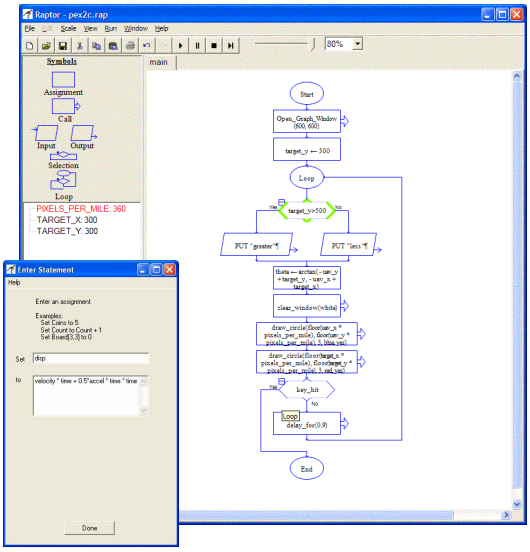
Tools to Introduce CT





Tools to Introduce CT

RAPTOR
Flowchart-Based
Programming
Environment



Tools to Introduce CT

ECS Exploring Computer Science

Tools to Introduce CT

BirdBrain TECHNOLOGIES LLC

Finch

ipre Scribbler

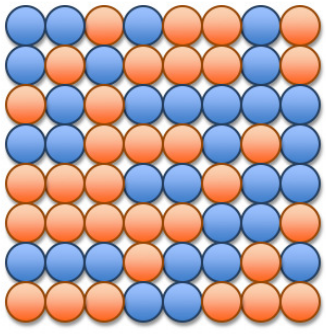
Tools to Introduce CT

COMPUTER SCIENCE Unplugged

Principles of Computing at CMU

- Introduce computational thinking principles.
- Use programming where appropriate to illustrate some of these ideas.
  - Introduce a minimal amount of syntax so students can focus on the computational principles, not on the syntax.
- Programming assignments include algorithms at various levels of detail.

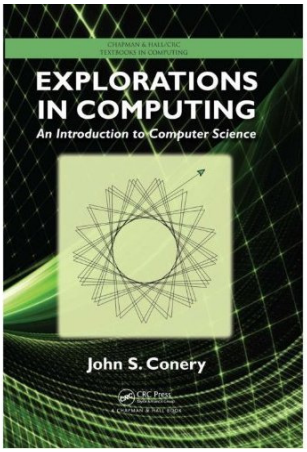
Course Logo

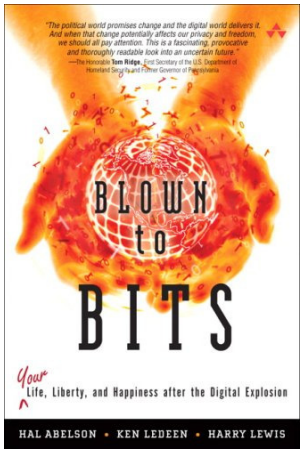


Class Statistics

- Fall 2010: 280 students
- Spring 2011: 380 students
- No CS or Computer Engineering majors
- All other majors represented:
  - Humanities, Business, Fine Arts, Science, Math, Engineering, Information Systems
- Lecture (3 hours/wk), Lab (1 hour/wk)
- 22 Assignments (problem sets, programming)
- 12 Labs
- 3 Written Exams, 2 Lab Exams, 1 Final Exam

Textbooks





Major Topics

- History of Computing
  - Babbage, Hollerith, von Neumann, Hopper, Moore, Turing, others
  - Census and wars cause huge leaps in computation
- Introduction to Ruby
  - variables, functions, assignments, for loop
- What is an Algorithm?
  - Lightbot, PBJ, GCD, Sieve of Eratosthenes
  - Ruby: while loop, if statement, array (list)

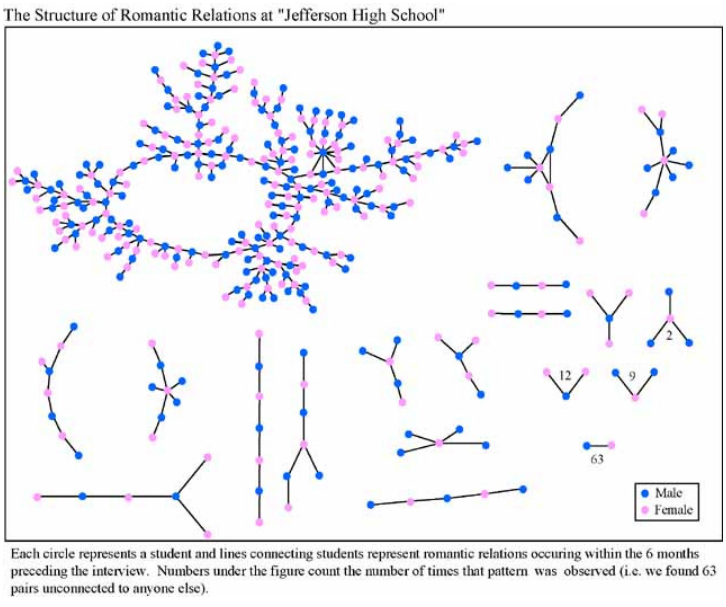
- Input: slices of bread, jar of peanut butter, jar of jelly

1. **Pick up some bread.**
2. **Put peanut butter on the bread.**
3. **Pick up some more bread.**
4. **Open the jar of jelly.**
5. **Spread the jelly on the bread.**
6. **Put the bread together to make your sandwich.**

- Output?

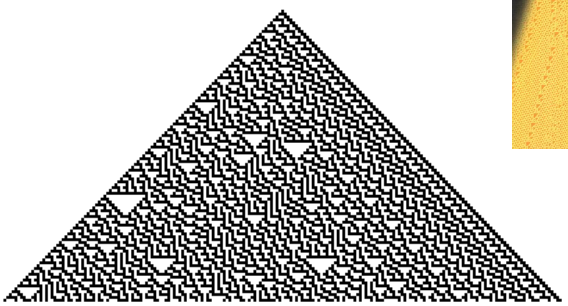
Major Topics

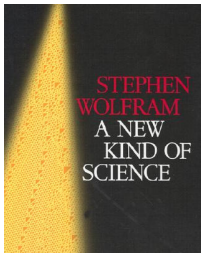
- Iteration
  - Linear search, insertion sort
  - Introduction to big O, loop invariant
- Recursion
  - Binary search, merge sort, more big O
  - Fractals
- Data Organization
  - arrays vs. linked lists, stacks and queues, hash tables, binary trees (BST and heaps), graphs

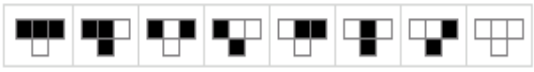


Major Topics

- Binary Representation
  - integers, floating point, characters, pixels, audio
  - compression: Huffman trees, JPEG, MP3
- Computer Organization
  - Gates, Boolean logic
  - Abstraction (Layers of a computer system)
- Randomness in Computation
  - Pseudo random number generators
  - Cellular Automata (Wolfram)

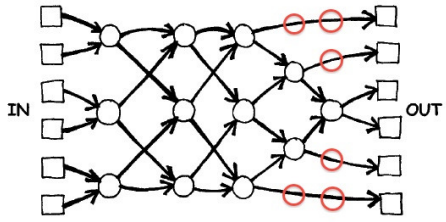


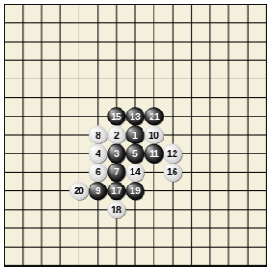




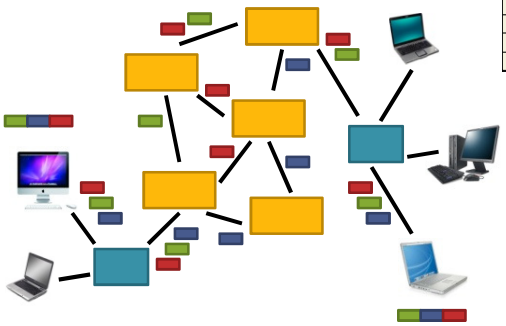
Major Topics

- Concurrency
  - multitasking, pipelining, sorting networks, deadlock, distributed computing
- The Internet
  - Addressing, Routing, Encryption
  - Abstraction: Communication Layers
- Artificial Intelligence
  - Natural Language Processing (Eliza, intro to regexp)
  - Game Trees and Heuristics
  - Machine Learning (Watson & Jeopardy!)

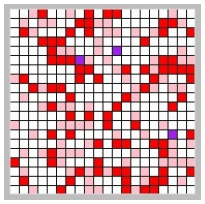




Gomuku



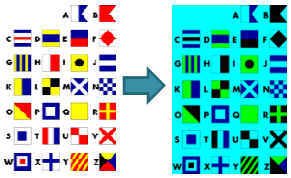
Major Topics

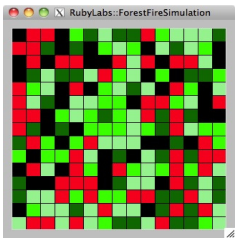


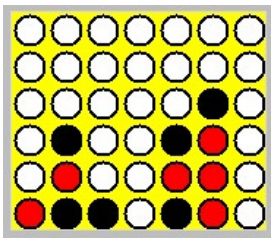
- Simulation
  - Principles, Flu virus simulation
- Limits of Computation
  - Intractability, more big O
  - P vs NP
  - Non-computability (Halting Problem)
- Future of Computing
  - DNA computing, Quantum computing, robots

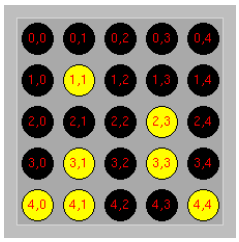
Some Programming Assignments

- Shortest Path
- Image Processing
- Forest Fire Simulation
- Games: Connect Four, Lights Out









Open Learning Initiative

- 3 modules for online learning:
  - Cellular Automata
  - Encryption
  - Computability
- Additional modules being developed this summer:
  - Iteration
  - Recursion
  - Binary Numbers

Summary

- A CS "crisis" makes the CS community examine why CS enrollment and interest has dropped.
- The phrase computational thinking captures the connections between CS and many fields.
  - CT argues for the need for computer science as part of K-12 general education.
  - Many tools exist to help illustrate CT ideas and an additional AP CS course is being developed to appeal to a wider audience.
  - Principles of Computing at CMU is one example of such a course.
- An introduction to CS doesn't have to be a course in computer programming!

Resources

www.cs.cmu.edu/~CompThink
www.csprinciples.org
www.iste.org/learn/computational-thinking.aspx
csta.acm.org/Curriculum/sub/CompThinking.html
www.alice.org
scratch.mit.edu
armorgames.com/play/6061/light-bot-20
raptor.martincarlisle.com
www.exploringcs.org
www.csunplugged.org
www.cs.cmu.edu/~tcortina/15110sp12
www.cmu.edu/oli