SECANT Workshop 2008
Teaching Key Principles of Computation without Programming

Dr. Tom Cortina
Assistant Teaching Professor
Computer Science Department
Carnegie Mellon University
How do we introduce non-majors (and budding majors) to CS?

- **CS0 Introduction to Computer Science**
  - Typically a crash-course in every computer science topic in one class.
  - Students are bombarded with one week of each major area of CS.
  - How much do they really absorb?

- **CS1 Computer Programming with $x$**
  (where $x = \text{Java, C, Scheme, C\#}$, etc.)
  - Students spend a great deal of time learning how to "speak" in this new language (syntax).
  - Different paradigms don't help matters.
  - Some CS1 courses use micro-worlds, multimedia, robotics, etc.
Principles of Computation (15-105)

- Survey of the major contributions and issues associated with computer science, focusing on the study of the process of computation.
  - CS is not viewed through a specific programming language.
  - CS is not viewed through a specific application area (e.g. multimedia, robotics, etc.)
- This course focuses on what it means to perform computation and what issues arise as mankind automates this process using computers.
- Designed for students who will probably take only one CS course in their lives.
Course Topics

• History of Computation
  * What societal needs caused mankind to make great advances in understanding and automating computation?
Course Topics

- Expressing Computation using Algorithms
  - Why do computations need to be expressed precisely?
  - How do we trace an algorithm to see what it is computing?
  - What common building blocks are used to specify algorithms?
• Our course uses a public-domain program called RAPTOR, a flowchart simulator.
• Students can build simple procedural programs without learning the syntax details of a language.
• Contains conditionals, loops, input and output, arrays, subroutines, graphics.

http://raptor.martincarlisle.com/
Programming without Syntax

Four In A Row

GAME OVER
PLAYER 1 WINS!

15 Puzzle

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>12</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>13</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>11</td>
<td>4</td>
</tr>
</tbody>
</table>

Lights Out

Each circle has a radius of 25 pixels.

Circles are either yellow (on) or white (off).

Pixel (1,1)
Pixel (250,250)
Course Topics

- Organizing Data
  - Are there better ways to **structure** data than just using a linear arrangement (vector)?
  - In what situations would we use these structures?
Course Topics

• Expressing Computations to a Computer
  - How do we use programming languages to express our algorithms to a computer so it can compute them?
  - Why are there so many programming languages?
  - Does the computer "understand" any of them?
"Tricks of the Trade" (Algorithmic Techniques)

- How do we express computations recursively? Is this more intuitive?
- How is "divide and conquer" used in computation?
- Does a greedy approach to solving a problem always give the optimal answer?
Course Topics

- Perfecting Computation
  - How do we know if a computation is correct? Why is this important?
  - How do we measure how efficient a computation is?
Course Topics

• The Limits of Computation
  - What makes a computation **intractable**?
  - Are all computations theoretically **solvable**?
  - Can we describe a **universal computer**?
Course Topics

- Parallel and Distributed Computation
  - What problems occur if multiple computational processes need to use the same resources? Can **deadlock** be avoided?
  - If we run a computation in **parallel**, how much faster do we get an answer?
Course Topics

• Applications
  ✷ What are the computational aspects of public-key, RSA encryption?
  ✷ How *intelligent* are computers anyway?

• Future of Computing (DNA and quantum computing)
Enrollment

- Enrollment

- Requirements
  - Technical Writing: Required as 1 course of a 2 course CS sequence.
  - Business: Acts as an alternative to Introduction to Programming.
  - Humanities/Social Science: Satisfies a Gen. Ed. requirement for modeling (mathematics and experiments).
Student Feedback 2006-2007

- Continue the use of Raptor?
  - Continue 30%
  - Continue but use more graphics 55%
  - Use another language/application 20%
  - Stay off the computer! 5%

- Did you take another CS class before or during this class?
  - Yes 20%  No 80%

- Are you interested in taking another CS class if you could?
  - Yes 55%  No 35%  Unsure 10%

- Would you recommend this course to your friends?
  - Yes 85%  No 15%
Feedback

- “I wish I had taken this course before my programming class. Now I understand what I was doing in Java.”
- "An alternative/substitute to [introductory programming]."
- "You learn a lot more than just how to code."
- "It's a lot better than doing real programming."
- "Understanding the world of computing beyond programming, and why and how programs and computers actually work."
- "It's a way to stay away from programming while actually learning something."
- "The fundamentals of programming.“ (?)
Computer Science Unplugged (K-12)

- CS Unplugged is a book of activities that illustrate computer science principles without using a computer.
- Activities are short and are designed to be easily integrated into classes and include exercises and lesson plans for teachers.

Created by Tim Bell, Ian H. Witten and Mike Fellows

Adapted for classroom use by Robyn Adams and Jane McKenzie
The basic edition of Computer Science Unplugged has 12 classroom exercises for you to use with your students.

Each exercise has a number of extensions, activities and background information.

All activities can be done without the use of computers, but they all demonstrate fundamental principles used in computers today.
Card Flip Magic
Beat The Clock
Treasure Hunt

What is the quickest route?

"directed graph"
Summary

• Computer science has revolutionized how scientists do science, businesses do business, …

• Students still see CS as just programming.

• Principles of Computation and CS Unplugged show the principles of computer science to all students without delving into complex programming syntax.
  - http://www.cs.cmu.edu/~tcortina/15-105sp08
  - http://www.csunplugged.org