CRA Conference at Snowbird 2006

Andy van Dam
Brown University
June 25-27, 2006
Programming is a Mode of Thought

• It isn’t Computer Science but a key component and a gateway to CS as well as Computational X

• Teaching students of all persuasions and interests how to be competent in programming is a “good thing”

• No claim that my “hard core” approach is the only approach or the best one – it works
CS15: Introduction to Object Oriented Programming

• The first of a two-course introductory sequence;
  - a smaller rival course teaches Scheme, ML and Java, more likely to be taken by those with some experience

• Assumes no prior background in CS
  - Welcomes, if not caters to, newbies

• Now about 100 students enrolled yearly
  - 30% female.
  - Fewer than 50% will be CS majors; many will take additional courses

• Attempts to teach OOP and software design through intense, immersive experience
  - Students work steadily

• A strong focus on interaction via GUIs
  - Great for OOP; keeps students more interested

• Followed by Algorithms and Data Structures (in Java)
CS15’s Approach to Teaching

• Teach objects first
  - Avoid inducing a hybrid procedural/OOP style of coding
  - All of OOP before many standard programming concepts
    - e.g. polymorphism before flow-of-control

• Learn by doing – lectures nearly irrelevant

• 8 substantial programs, no exams, quizzes
  - Including Tetris and a large final project
  - Final projects reach several thousand lines
  - All programs have written design elements which must be handed in before the final program is due

• CS15 makes heavy use of pre-written libraries
  - So called “magic” is inevitable
  - Better to learn how to use them
  - Students do learn lower level concepts in the third course taken by CS concentrators at Brown
CS15’s textbook

Object-Oriented Programming in Java
A Graphical Approach

Kathryn E. Sanders & Andries van Dam
Methodology

• Detailed slide sets for every lecture
  - use PowerPoint® animations to illustrate concepts visually
  - Posted on the course website
  - Lectures are recorded with both video of the slides and audio of the lecture to further encourage review

• Java demos to immediately demonstrate uses for concepts

• Large staff of Undergraduate Teaching Assistants (UTAs) 1UTA/8 students
  - Allow for 60+ office hours per week
  - UTAs lead help sessions for every program to go over high-level design concepts and to answer questions about support code, requirements, etc.
  - Provide detailed feedback on design decisions in both written design hand-ins and programs

• Introduce the excitement of CS with short show-and-tell by other profs, cameos by former students, typically female
Sample CS15 Slide:
Location/Dimension

• The screen is a grid of **pixels** (tiny dots)
  - “picture elements”

![Diagram of pixel art and bounding box](image)

• Unlike a Cartesian plane!
  - the **origin** is in the **upper-left corner**
  - the **y-axis increases downward**

• The **location** of any shape is described by
  the **upper-left corner** of its bounding box
Sample CS15 Slide: Repaint!

```java
JPanel

paintComponent(Graphics2D brush) {
    super.paintComponent(g);
    Graphics2D brush = (Graphics2D) g;
    _rectangle.paint(brush);
}

drawComponent(Graphics brush) {
    brush.setColor(_borderColor);
    brush.draw(_shape);
    brush.setColor(_fillColor);
    brush.fill(_shape);
}

canvas.repaint()
```

Someone

drawComponent(Graphics brush) {
    brush.setColor(_borderColor);
    brush.draw(_shape);
    brush.setColor(_fillColor);
    brush.fill(_shape);
}

canvas.repaint()
Panel Questions (rephrased)

• Best approach?
  - Whatever prof is passionate about

• How to get more suckers into the tent?
  - Game design, Alice and other forms of much more instant gratification, Digital Visual Literacy, …

• Does approach scale?
  - Yes. Need UTAs, vanilla machines

• Is lack of experience an inhibitor?
  - No evidence at Brown, and I prefer newbies

• Turn-off factors?
  - Pace/intensity
  - Lack of collaboration
  - Lack of real-world applicability – would be great if in an intro course you could solve a societal problem in well-defined steps, in synch with the machinery being taught