Computer Science for the Masses

Robert Sedgewick
Princeton University

[ joint work with Kevin Wayne ]
This talk is dedicated to the memory of Philippe Flajolet

Philippe Flajolet 1948–2011
Computer Science for the Masses

- Mission accomplished?
- Disruptive changes
- A way forward
- Taking the plunge
- Postscript
Central thesis for intro CS (1992)
First-year college students (and high school students) need a computer science course

Computer science embraces a significant body of knowledge that is
• intellectually challenging
• pervasive in modern life
• critical to modern science and engineering

Anyone can learn the importance of
• modern programming models
• the scientific method in understanding program behavior
• fundamental precepts of computer science
• computation in a broad variety of applications
• preparing for a lifetime of engaging with computation

Goal: A standard intro text for CS that can stand alongside standard intro texts for other fields.
Basic approach

Fact: Many *scientists* lack basic knowledge of *computer science*.

Fact: Many *computer scientists* lack basic knowledge of *science*.

1970s: *Anyone using the computer* needs intro CS.

1990s: *Future cubicle-dwellers* need intro CS.

2010s: *Everyone* needs intro CS.

To address the situation:
• Identify the fundamentals.
• Teach them to *everyone*.
• Do it *as early as possible*. 
Our introductory CS course model

Computer Science for the masses

Original motivating factors (1992)
• Why not?
• Works for biology, math, physics, economics.
• It is our responsibility (if not CS faculty, who?).

*are more important than ever.*
[20+ years of difficult challenges omitted.]

- Identify content
- Change content
- Interface with the computer center
- Choose programming language
- Dot-com bust (enrolls way down)
- Change programming language
- Staffing
- Political battles
- Competing courses
- Inadequate resources
- Abandon computer center
- Financial crash (enrolls way up)
- Windows, OS X, Linux
- ...
What is this course about?

A broad introduction to **computer science**.

**Goals**
- Demystify computer systems.
- Empower you to exploit available technology.
- Build awareness of substantial intellectual underpinnings.

**Topics**
- **Programming** in Java.
- **Design** and architecture of computers.
- **Theory** of computation.
- **Applications** in science and engineering.

"Science is everything we understand well enough to explain to a computer."

---

"Computers are incredibly fast, accurate, and stupid; humans are incredibly slow, inaccurate, and brilliant; together they are powerful beyond imagination."

---

– Don Knuth

– Albert Einstein

and art, music, finance, and many other fields.
Outline of course content

**Computer Science**  
R. Sedgewick and K. Wayne

**Elements of Programming**  
Your First Program
Built-in types of Data
Conditionals and Loops
Arrays
Input and Output
Case Study: Random WebSurfer

**Functions and Modules**  
Static Methods
Libraries and Clients
Recursion
Case Study: Percolation

**Object-Oriented Programming**  
Data Types
Creating DataTypes
Designing Data Types
Case Study: N-body

**Algorithms and Data Structures**  
Performance
Sorting and Searching
Stacks and Queues
Symbol Tables
Case Study: Small World

**Theory of Computation**  
Formal Languages
Turing Machines
Universality
Computability
Intractability

**A Computing Machine**  
Data representations
TOY machine
Instruction Set
Machine-Language Programming
Simulator

**Building a Computer**  
Basic Circuit Model
Combinational Circuits
Sequential Circuits
Digital Devices
All in the context of applications

Ideal programming example/assignment

• teaches a basic CS concept
• solves an important problem
• is intellectually engaging and appealing
• is open-ended

```java
public class BouncingBall {
    public static void main(String[] args) {
        // Simulate the movement of a bouncing ball.
        double rx = .480, ry = .860;
        double vx = .015, vy = .023;
        double radius = .05;

        StdDraw.setXscale(-1.0, +1.0);
        StdDraw.setYscale(-1.0, +1.0);

        while(true) {
            // Update ball position and draw it there.
            if (Math.abs(rx + vx) + radius > 1.0) vx = -vx;
            if (Math.abs(ry + vy) + radius > 1.0) vy = -vy;

            rx = rx + vx;
            ry = ry + vy;

            StdDraw.clear();
            StdDraw.setPenColor(StdDraw.BLACK);
            StdDraw.filledCircle(rx, ry, radius);
            StdDraw.show(20);
        }
    }
}
```
2011: Time to declare victory?

Introduction to CS enrollments

- **Double** the height of the “bubble”
- 43% of all Princeton students.

“Algorithms” enrollments

- **Three times** the height of the “bubble”
- 17% of all Princeton students.

Q. (2011) When will enrollments start to decline?
Mission accomplished? (2011)

RS. *Algorithms for the masses* (ANALCO, San Francisco 2011)

**Summary**

The scientific method is an essential ingredient in programming. Embracing, supporting, and leveraging science in intro CS and algorithms courses can serve large numbers of students.

**Proof of concept:** First-year courses at Princeton

- 40+% of Princeton students in a single intro course
- 25+% of Princeton students in a single algorithms course

**Next goals:**

- 90+% of all college students in an intro CS course
- 50+% of all college students in an algorithms course

*Computer Science for the masses*

Confession: No idea how we would get there...
Computer Science for the Masses

• Mission accomplished?
• Disruptive changes
• A way forward
• Taking the plunge
• Postscript
Seismic changes are afoot

For a millennium, universities have been considered the main societal hub for knowledge and learning. And for a millennium, the basic structures of how universities produce and disseminate knowledge and evaluate students have survived intact...Today, though, the business of higher education seems to some as susceptible to tech disruption as other information-centric industries


Business of higher education ?? A road to ruin.
Sit in your local coffee shop, and your laptop can tell you a lot. If you want deeper, more local knowledge, you will have to take the narrower path that leads between the lions and up the stairs. There—as in great libraries around the world—you’ll use all the new sources, the library’s and those it buys from others, all the time. You’ll check musicians’ names and dates at Grove Music Online, read Marlowe’s “Doctor Faustus” on Early English Books Online, or decipher Civil War documents on Valley of the Shadow. But these streams of data, rich as they are, will illuminate, rather than eliminate, books and prints and manuscripts that only the library can put in front of you.

The narrow path still leads, as it must, to crowded public rooms where the sunlight gleams on varnished tables, and knowledge is embodied in millions of dusty, crumbling, smelly, irreplaceable documents and books.
While Grafton’s reservations about putting knowledge online are well taken, I would also point out that there is quite a bit going on now in the academic world that doesn’t have much to do with old books. Indeed, as the author of many books, I wonder whether perhaps the book is not quite sacred as a means of disseminating knowledge.

What is the most effective way to produce and disseminate knowledge with today’s technology? How can we best structure what we know and learn so that students, researchers, and scholars of the future can best understand the work of today’s researchers and scholars?

I think that questions like these are more important and more difficult to address than whether we can put the contents of libraries on the Web.
Future of libraries?

1980s
- Students spend significant time in the library
- Faculty members depend on the library for research

2010s
- Students spend significant time online and have *no need* for the library
- Few faculty members in the sciences use the library *at all* for research

2020s?
- A few *book museums* (for Grafton)
- Digital library infrastructure (for everyone else)

How will we disseminate knowledge in the future?
Will universities play a role?
Disruptive change II: Textbooks

We are on a road to ruin

- Prices continue to escalate.
- Students now *rent*, not own books.
- Planned obsolescence? Walled gardens?

Princeton U-store
1950s

Princeton U-store
2010s

Is there room for a good textbook?
Will free web resources prevail?

No books!
Disruptive change III
2012: MOOCs go mainstream

Q. (Jan. 2012) Are you interested in teaching online?
RS+KW. No. (Too much work to do it properly.)

Q. (Apr. 2012) Trustees want it: we're doing it anyway. Are you in?
RS+KW. An offer we cannot refuse...

With apologies to our actual administrators

An online platform for the "course" abstraction

"Algorithms, Part I" (Summer 2012)

Immediate realization:
Our model is perfectly suited to go online.

Andrew Ng and Daphne Koller
[4 years of difficult challenges omitted.]

- Production design
- Record large lectures?
- Which presentation software?
- Developing assessments
- Can we do math this way?
- Who pays?
- Who owns it?
- “You’re just a troublemaker”
- Crashing the Amazon cloud
- Builds in the presentations
- Platform issues
- Lawyers and contracts
- ...

Brief summary of MOOC experience

Facts and figures

• Six courses produced, four already deployed.
• 70+ lectures, each running 60-90 minutes.
• 3000+ state-of-the-art lecture slides.
• Over 1 million people reached.

Distribution model

• Courses offered twice/year as Coursera MOOCs.
• Each course has an associated textbook.
• Lecture videos also bundled with the textbooks.
• Each textbook has an associated booksite.
FAQs

Q. Isn't developing an online course time-consuming and difficult?
A. Yes! 50-100 hours of preparation per lecture.
A. Yes! My workflow requires skill in Unix, emacs, Mac OS X, Dropbox, Illustrator, Indesign, Java, C, TeX, video capture, KeyNote, Acrobat, HTML, MathJax, Mathematica, PostScript, and a dozen other tools I can't even name.
A. It's less time-consuming than writing a book.

Q. I think my administration will fire me and use your lectures.
A. I'm wondering about that myself!
A. Cut costs by firing teachers? I think not.
A. You can use a blended model, where students watch my lectures and you make sure they understand the material and you evaluate your students.
A. Does it make sense for all of us to be preparing and delivering lectures on Quicksort, hashing, and a dozen other topics every semester? Just use good online lectures.
A. Your students are already watching my lectures.
Computer Science for the Masses

- Mission accomplished?
- Disruptive changes
- A way forward
- Taking the plunge
- Postscript
A way forward

embraces technology to integrate three abstractions that are *here to stay*:

**“Textbook of the future” model**

- Studio-produced **video lectures** that *introduce* material and *inspire* further study.
- **Web repository** for use by students to *explore* and *interact* with the content.
- Authoritative **textbook** for use by students to *learn* and *study* the material.
The “lecture” abstraction
has been an essential part of education for a millenium and is here to stay

Advantages.
• Allows instructor to precisely control pace and direction.
• Stimulates development of a “community of scholars”.
• Encourages great teachers to inspire large groups of students.

Disadvantages
• Requires significant time and effort for preparation.
• Places students in a passive role.
• Requires instructor to have effective writing and speaking skills.

Is there a practical alternative?

Source: Office of Instructional Resources, University of Illinois-Urbana Champaign.

“Lecturing is that mysterious process by means of which the contents of the note-book of the professor are transferred … to the note-book of the student without passing through the mind of either.”

– Edwin Emery Slosson
Is there a practical alternative?

20th Century

21st Century
Lecture presentation materials are evolving to new standard of excellence.

Chalktalk

"Powerpoints"

This is a horse.

Overhead projection

State of the art presentations

Connected components in mappings

<table>
<thead>
<tr>
<th>Class</th>
<th>Y, the class of cycles of Cayley trees</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>EGF</td>
<td>[ Y(z) = \sum_{x \in Y} z^{#(x)} = \sum_{N \geq 0} \frac{Z^N}{N!} ]</td>
<td>1 10 2 1 10 2 9 1 2 3 1</td>
</tr>
</tbody>
</table>

Construction

EGF equation

Extract coefficients by Lagrange inversion with \( f(u) = u^6 \) and \( \mu(u) = \ln(1/1-u) \)

\[
Y_N = N! [z^N] Y(z) = N^{N-1} \sum_{1 \leq k < N} \frac{N!}{N^k (N-k)!} = N^{N-1} Q(N) \sim \frac{N^N \sqrt{\pi}}{2N}
\]

Top-down s-3-4 trees

Transform tree on the way down to ensure that last node is not a 4-node.

"Splits" any 4-nodes at one of two trans.

Invariance: "current" node is not a 4-node insertion at bottom therefore easy.

1 2 3 1 2 3 1

Lagrange Inversion Theorem (Brauer's form)

If a GGF \( g(y) = \sum_{k \geq 1} c_k y^k \) satisfies the equation \( y = f(e^{-y}) \) with \( f(0) = 0 \) and \( f'(1) = 0 \) then, for any function \( f(x) \),

\[
[M(x)] f(y) = \frac{1}{1 - e^{-y}} \sum_{k \geq 0} \left( \frac{y}{k!} \right)^k f(k) e^{ky}
\]
### Mathematical derivations

**Cayley trees**

<table>
<thead>
<tr>
<th>Class</th>
<th>$\mathcal{C}$, the class of labeled rooted unordered trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>EGF</td>
<td>$C(z) = \sum_{c \in \mathcal{C}} \frac{z^{</td>
</tr>
</tbody>
</table>

**Example**

```
6 2 1 1 2 2 5 1
```

### Plot

- Points of Cauchy function
- $\psi_0 = 0$
- $\phi_{1/2} = 1$

### Drawings of combinatorial objects

#### Classic example of the symbolic method

**Q.** How many trees with $N$ nodes?

- $G_1 = 1$
- $G_2 = 1$
- $G_3 = 2$
- $G_4 = 5$
- $G_{14}$

- "Builds" control pace.
- Details to support reasoning are included.

---

**Story**

Analytic combinatorics overview

- **A. SYMBOLIC METHOD**
  1. OGFs
  2. EGFs
  3. MCFs
- **B. COMPLEX ASYMPTOTICS**
  4. Rational & Meromorphic
  5. Applications of R&M
  6. Singularity Analysis
  7. Applications of SA
  8. Saddle point
Online studio-produced lectures

transform lectures from passive to active learning experiences for students

Students actively choose their own pace
- Typical beginners slow the pace at first.
- Typical advanced students view lectures at double speed.
- Everyone’s pace varies throughout the course.

Students actively choose the time and place they learn
- “Last thing in the evening, lying in bed.”
- “First thing in the morning, in the library.”
- “On the team bus.”

Lectures are always available for review
- Students review the material until they understand it.
- “Office hours” are dramatically reduced.
- Exam review is much less stressful.
The "textbook" abstraction has been an essential component in education for centuries and is here to stay.

Well-understood since the Greeks.

Enabled for the masses by Gutenberg.

Advantages

- Articulates what students can reasonably learn about a subject in a semester.
- Distills a lifetime of faculty experience for future generations.
- Provides a reference point for future studies related to the subject.

Disadvantages

- Need to be written by professors.
- Extremely difficult to produce.
- Publishing industry imploding.
Publishing technology is empowering individuals to be more productive than ever before.
The "web content" abstraction is emerging as an essential component in education and is *here to stay*.

What is web content?

- *Full coverage* integrated with *web search*.
- Always up to date (*dynamic*).
- Content types *not available* in print.

**Issues**

- Basic properties still evolving.
- Free? Who pays?
- Who creates it? Who maintains it?
Proof of concept

embraces technology to integrate three abstractions that are *here to stay*:


- Textbook in print in various versions since 1982, 700,000+ copies sold.
- Web repository developed by Kevin Wayne in mid 2000s.
- Studio-produced videos were in the first wave of MOOCs
Coming in 2016

*Computer Science “Textbook of the future”*

- Studio-produced *video lectures* that *introduce* material and *inspire* further study.
- *Web repository* “booksite” concept developed by Kevin Wayne in early 2000s.
- New *textbook* to be published in the early summer.

---

**introcs.cs.princeton.edu**

- 10000+ files
- 2000+ Java programs
- 50+ animated demos
- 2 million+ visits/year
Computer Science for the Masses

- Mission accomplished?
- Disruptive changes
- A way forward
- Taking the plunge
- Postscript
Sudden realization (spring 2015)

20th Century

Exactly how are we going to be teaching computer science *at Princeton* in the future?

21st Century
How are we going to teach CS in the future?

Goals

• Continue to expand, to serve all students.
• Continue to improve learning outcomes.

Implication: Need a scalable model.

Tactics

• *Exploit technology to the extent possible.*
• Improve things that are working.
• Throw out things that are *not* working.

RS: Hey, we *have* to use the studio-produced lectures!
Everyone else: *Why would we change our biggest and best course?*
[6 months of difficult negotiations omitted.]

- Students won’t watch
- Rules won’t permit it
- Will require preparation of new material
- Who will teach it
- How do we change videos
- Video editing
- Who can watch them?
- Staff will need to reteach
- System won’t support it
- Too hard to set up
- ...

...
Glitches (not unusual)

- Over 90 degrees in the room.
- Biggest lecture hall on campus is too small.
- Students in aisles cannot see the screen.
- Sound system stops working halfway through.

Consequence. All students motivated to move online!
An unqualified success

Q. What do you think of the online lectures?

A. 82% of responses were positive.

Students *loved* active participation in consuming lecture content

- “Prepares me for a lifetime of active learning online.”
- “I like this system, it really lets me go at my own pace and rewatch if I need to.”
- “The video lectures are *amazing*. I believe many classes would benefit from this.”

Course staff also reaped benefits

- No need to reteach lecture material in office hours.
- More time for interaction with students in small groups.
- More time for interaction in large class meetings.
- Scheduling complications virtually eliminated.
2015: Time to declare victory? Not yet!

Introduction to CS enrollments

- *Triple* the height of the “bubble”
- 2/3 of all Princeton students.
- Largest course at Princeton

"Algorithms" enrollments

- *Four times* the height of the “bubble”.
- *Doubled* in the last two years.
- 40% of all Princeton students.
- 4th largest course at Princeton

Q. (2011) When will enrollments start to decline?

A. (2015) Sometime after they stop accelerating!
Primary advantage of “textbook of the future" model

is scalability allowing reach orders of magnitude more students

Book, course, and booksite are 21st century resources. Intended audience: Teachers, students, and anyone wanting to learn CS.
Scalability plus “CS for everyone” approach promotes *diversity* because everyone is prepared for further study in CS.

**Bottom Line.** Nearly 40% women majors, *more than twice* the national average.
Computer Science for the Masses

- Mission accomplished?
- Disruptive changes
- A way forward
- Taking the plunge
- Postscript
Q. Isn't developing an online course time-consuming and difficult?
A. Yes! 50-100 hours of preparation per lecture.
A. Yes! My workflow requires skill in Unix, emacs, Mac OS X, DropBox, Illustrator, InDesign, Java, C, TeX, video capture, KeyNote, Acrobat, HTML, MathJax, Mathematica, PostScript, and a dozen other tools I can't even name.
A. It's less time-consuming than writing a book.

Q. I think my administration will fire me and use your lectures.
A. I'm wondering about that myself!
A. Cut costs by firing teachers? I think not.
A. You can use a blended model, where students watch my lectures and you make sure they understand the material and you evaluate your students.
A. Does it make sense for all of us to be preparing and delivering lectures on Quicksort, hashing, and a dozen other topics every semester? Just use mine.
A. Your students are already watching my lectures.
Questions answered (administrators)

Q. Why aren't more faculty developing online courses?
A. You are providing only a tiny fraction of the resources needed.

Q. What can we do to improve things?
A. Embrace technology.
A. Develop a teaching class of professors who can teach blended classes.
A. Provide support and real incentives for teaching and content creation.
A. Invest in research at the interface of education and technology.
A. Attract and provide resources to the best and brightest professors in the field.
A. Develop academic leadership for the effort.

Q. How much will it cost our institution to embrace online education?
A. Less than you are spending on many things that are less central to your mission.
A. You need to plan to invest in this at the scale you are investing in the library.
A. Can you afford to not embrace online education?
A parting thought

(from John Hennessy in an interview for an article by Ken Auletta the New Yorker,

“[Universities,] like newspapers and music companies and much of traditional media a little more than a decade ago are sailing in seemingly placid waters.”

“But ... there’s a tsunami coming.”
What happened to the tsunami?

I think the bloom is now off the rose, and now is going to be the time when some really hard-nosed thinking has to be done about the true value of these online courses.

Shirley Tilghman, 2013

Stumbling blocks

- *Institutions* are trying to take control (and failing).
- Misplaced focus on credentials and degrees.
- Content creation is the province of *individuals*.
- Bad business models, created prematurely.

Result: Plenty of money sitting on the table.

**RS: Looks like a tsunami to me!**

- Tens of thousands of pages of online content
- 100+ hours of lecture videos.
- Reaching millions of individuals.
- 1990s: Lucky to be able to teach my own children.
- 2030s: Will be teaching my own *grandchildren*. 
Computer Science for the Masses

Robert Sedgewick
Princeton University

[ joint work with Kevin Wayne ]