UW’s CSE120 Computer Science Principles
Lawrence Snyder
For years there have been non-majors CS courses ... often just schlock branded ‘literacy’
A key question: Should non-majors learn programming ... argument raged for decades
Eventually ...
Bill Wulf, Assistant Director of CISE at NSF, requested a study on the topic “What should everyone know about computer technology?” ... 1997

Different situation than other studies
  - Field had matured – we know CS is about
    - algorithms, information, communication, process ...
  - Public sensed that computers were powerful and could do more for them
  - Many academics were finally applying computers in their disciplines
NRC Has Visibility

- The “Fluency Study” got national visibility
- Committee focused on fundamentals
  - Al Aho
  - Marcia Linn
  - Arnie Packer
  - Larry Snyder (Chair)
  - Alan Tucker
  - Jeff Ullman
Fluency ≠ Literacy 2.0

Fluency is composed of 3 parts, topics listed (10)

- **Skills** = Power Users Today; will change over time search, directory hierarchy, a power app or 2, ...
- **Concepts** = Basic Science; will not change bits, data, algorithms, programming, DB, networks, ...
- **Capabilities** = Higher-level Thinking; always helps logical thinking, debugging, authoritativeness, ...

Report focused on college; better if learned before
In Fluency, programming is taught as a means of teaching algorithms, and other topics.

When expertise isn’t the goal, less is needed.

It can be simplified, but programming cannot be made trivial, so inevitably the class will require students to work and think ...

... but they will think algorithmically.

This formulation ended the “programming for non-majors?” debate.
As part of her Broadening Participation in CS program, NSF-er Jan Cuny funded a study to create a new AP Computer Science Principles course.

- AP brand has many advantages
- The Advisory Commission formulated 7 Big Ideas and 5 Computational Thinking Practices
7 Big Ideas

- Computing is a creative activity engendering innovation
- Abstraction reduces detail to focus on concepts
- Data & information facilitate the creation of knowledge
- Algorithms are tools for expressing solutions
- Programming is a creative process implementing human intent
- Digital devices, systems and networks foster computational approaches to problems
- Computing has impact, enabling advancement
7 Ideas + 5 Practices ≠ Syllabus

- Commission’s strategy to develop course: pilots
- 5 professors developed a class for their campus

Tiffany Barns  
UNC Charlotte  
Beauty & Joy of Computing

Dan Garcia  
UC Berkeley  
Beauty & Joy of Computing

Jody Paul  
Met St College  
Living In A Computing World

Beth Simon  
UC San Diego  
Fluency with Information Technology

Larry Snyder  
U Dub  
Computer Science Principles
Balance of Topics

- UW compared to Combined
The five classes were studied in depth to verify

- They were interesting to all students

Student Engagement

- Lecture (N=37)
- Discussion (N=32)
- Labs/Sections (N=26)
- Projects/Problem Sets (N=11)
- Group Projects (N=15)
Pilot class became UW’s CSE120 CS Principles

Available since 2011 via UW In The High Schools

First adopters:
- Susan Evans, Cleveland High
- Carl Walther, Sunnyside High

Course has morphed to be more effective at dealing w/ difficult concepts ... It’s now perfect!
Here’s Looking At You ...

- Taking a quick tour of the course approach and content ...
CSE120 is a CSE course, but not part of major

- 5 credits = 3 lectures + 2 closed labs over 10 weeks
- 1 hour of homework/day, MT, Final, pair pgmming
- Mostly freshmen, not intending CS or STEM majors

Available for UW in the High Schools

- College credit at UW; can be taught w/o UW role
- Best for year long course – taught in WA + other sts

Teacher resources

- ALL UW materials, lectures, lab, HW, exams, files
- 1 day teacher orientation to content & technique
Teaches 7 Big Ideas + Comp Thinking Practices
Programming is ~40% of instruction
Languages:
- Lightbot 2.0 – drag and drop game for week 1
- Processing Visualization Language – stealth lang
  - Not a toy; Java + IDE; segues to Java for CSE or AP CS-A
- XML plus needed XSL and HTML for DB
Topics: creativity, abstraction, functions, bits, parameters, privacy, security, encryption, RGB, complexity, databases, binary math, analog, compression, networking, DNS, AI, design, …
All of the scheduling information -- days off, assignments, due dates, reading links, etc. -- are presented on this page. Notice that some links are present, but not populated with a file yet.

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>Assignment</th>
<th>Due Today</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 5</td>
<td>Lec: Introduction, Expectations, The Plan</td>
<td>Slides</td>
<td>Assignment 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Precourse Survey</td>
</tr>
<tr>
<td>Jan 6</td>
<td>Lab: &quot;What I value&quot; writing exercise; FTP</td>
<td>Lab 1</td>
<td>Essay Collect</td>
</tr>
<tr>
<td>Jan 7</td>
<td>Lec: Lightbot 2.0 - A Game or Programming?</td>
<td>Slides</td>
<td>Assignment 2</td>
</tr>
<tr>
<td>Jan 8</td>
<td>Lab: Informal Algorithms</td>
<td>Lab 2</td>
<td>Bauby Blinks Collect</td>
</tr>
<tr>
<td>Jan 9</td>
<td>Lec: Functions -- the Punchline</td>
<td>Slides</td>
<td>Assignment 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Assignment 2 Collect</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>After Image Survey by 11:00 PM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Collect</td>
</tr>
<tr>
<td>Jan 12</td>
<td>Lec: Processing Introduction</td>
<td>Slides</td>
<td>Assignment 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Assignment 3 Collect</td>
</tr>
<tr>
<td>Jan 13</td>
<td>Lab: Writing First Processing Programs</td>
<td>Lab 3</td>
<td>Collect</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Robot Code</td>
<td></td>
</tr>
</tbody>
</table>
Design Criterion: Engage All

- Pilot goal – engage everyone; teach concepts
- CSE120 content carefully selected, tested
Balance Required

- Teaching AP CSP content is not the problem

Problem –

- Content must be interesting to keep at it OR add
  - Drawing Simpsons ➔ Avengers
  - Challenges and extra credit
- Empower All Students
  - No one is advantaged by content
  - Contain ‘hot shots’ by bringing all students along
  - In abstractions, try to limit “abstractness” ... Processing helps thanks to tight connection to images
  - Cycle through programming ideas multiple times
CSE120 is a science class teaching fundamental ideas of Computer Science
- We don’t use test tubes – CS is not ‘physical’
- We do teach the laws of nature concerning data, information, algorithms, abstraction, logic, etc.

This class is for general audiences
- It is not a “majors” class
- It could prepare you to take “majors” classes
Orientation ... the First Day

- Establish content
- Generate Interest
- Build Respect – both ways
- What’s your favorite food ...
This Course Covers 2 Kinds of Info

- I see the task of this course as teaching
  - Computational Principles – “bits can represent all information” – that everyone should know
  - Computational Thinking – ways you can use computers to solve (your) problems
I see the task of this course as teaching

- **Computational Principles** – “bits can represent all information” – that everyone should know
- **Computational Thinking** – ways you can use computers to solve (your) problems

If you were thinking this class will be ...

- **Trivial**, forget it: I teach stuff you haven’t had before
- **Difficult**, forget it: The class was designed to be a high school AP class
- **Fun and interesting**: Perfect ... that’s what it is
Some people panic at the mention of the word *programming* ... as if just saying it would cause them to become social outcasts, nerdy, ...
Some people panic at the mention of the word *programming* ... as if just saying it would cause them to become social outcasts, nerdy, ...

Programming’s a career; it takes years to learn; it pays really well; “normal people” do it, too

I teach some programming in this class as part of teaching computational thinking

- You won’t be a programmer at the end
- You will still be however “normal” as you are today
- You will, I hope, also think differently as a result
**Class Structure**

- 3 lectures – I will talk, demo and we’ll all discuss various topics
- 2 labs – practice with the TAs present
- Homework – exercises that help push the material further ... *one hour a day, as needed*

- “Team project,” Midterm, Final
- Fridays “After Image Survey” ... worth a few points, takes less than 5 minutes, gives me feedback on how things are going
Instructor: Larry Snyder, Prof. Emeritus
- I like travel; I’ve lived in > ½ dozen other countries
- A favorite food is anchovies + potato chips

TAs: Dun-Yu Hsiao, Sona Grigoryan, Bran Hagger, Geoff Liu

... and you?
- A Favorite Food
Starting out programming is tough, esp. for students who aren’t necessarily quantitative

What to do ...

- Many schools pick a drag/drop language, Scratch or Alice or similar
- This decision is problematic
- I prefer
  - Lightbot 2.0
  - Processing
What are you doing in Lightbot?

- Commanding a robot through a “blocks world”
- Programming is **commanding** an agent
*Sequencing*

- Instructions are *given* in sequence, i.e. in order
- They are *followed* in sequence, i.e. in order
  - YOU give the instructions ... it’s called *programming*
  - The AGENT follows them ... it’s called *executing* or *running* the program
  - A *program counter* marks the agent’s place
Yet Another Solution

Just Do It!
*Add Some Code*

- Type in instructions that you will learn shortly

Then run your program

```java
void setup() {
    size(500,500);    //define canvas size
    background(0,0,255);    //define canvas color
    stroke(0,0,255);      //define line color
}

void draw() {
    line(150,150, mouseX, mouseY);    //draw line from
    if (mousePressed){    //if the mouse is ever close
        stroke(255);
    }
}
```
What You See

- When you start up the Processing system...

![Programming window with buttons: Run, Stop, New, Open, Save, Export, and file name input field.]
Looking At Simpler Code

- Drawing a snow angel is straightforward ...

```java
void setup () {
    size(500,500);
    stroke(255,255,255);
    background(0,0,255);
}

void draw () {
    line(150,150,mouseX, mouseY);
}
```

Just Do It!
Look At Some Programs

- Lightbot 2.0
- Snow Angel
- Simpsons
- Simpsons 2
- Owls
- Mike
Five Years Experience

- Originally developed as Pilot for AP CSP
  - “In the 5 pilot class no statistical difference between how women and under-represented groups scored”
- 5 year record of attracting women + minorities
  - Over 5 years, in only 2 were more men than women
  - Half of students go on to take more “computing”
  - Women score well – in 2015 top 10% were women
  - Minorities – in 4 of 5 years African Americans and Native Americans both exceed campus average

*Early Assessment of CSE120 LS, RA-- forthcoming*