INTEGRATION

Are the people in our class students or are they learners?

How do you grab student attention on the first day and every day after?

Do you have a plan for conquering negative expectations?
Chrestomathics + Student-Focused Approaches = Success

Jim Cohoon
University of Virginia
Why you should care?

- It works
  - Connects educational strategies and practices in Lighthouse CC professional development to classroom success
  - Motivates student interest
  - Attracts students to follow on courses
  - Produces favorable outcomes for all types of students
What is going on here?
What is going on here?
What is Chrestomathics

- Study of interesting, useful things and processes
  - From problem to solution

- CS 1
  - Digital problem solving
Topics of interests
Topics of interests

```python
# purpose: demonstrate string acquisition (the getting of input)
# get input
reply = input('Tell me what is on your mind: ')
# echo reply
print('That is interesting. ')
print('Why is "" + reply + "" on your mind?')
```
Topics of interests

```python
# purpose: dateability app
# prompt user to supply an age
reply = input( 'Enter your age: ' )

# convert user reply to integer age
age = int( reply )

# compute min dateability age according to our folk rule:
#   you should only date someone who is at least half your
#   age + 7 years
dateable_age = ( age // 2 ) + 7

# print the result of the computation
print( age, 'year olds should only date someone' )
print( 'who is at least', dateable_age, 'years old' )
```
Topics of interests
What’s interesting and to who

Computing Interests Survey

Offering students interesting and meaningful examples that illustrate and apply computing concepts can improve learning. Your answers to the following optional and anonymous survey will help computer science instructors accomplish that goal.

Please rate your interest in the following topics as possible class and homework examples on a scale from extremely uninterested to extremely interested.

Information about a topic can be displayed by rolling the mouse over the topic.
Pathways
Climate change

- The plan
  - Enact proven *effective* changes that would work for *all* students

- Looked around
  - Joanne Cohoon, *Toward Improving Female Retention in the Computer Science Major*, CACM
First steps toward change

- Changes to CS1
  - Instructors
  - Labs
  - Programming language

- Outcomes
  - Student satisfaction with more majors
  - Increase in diversity
Next steps toward change

■ Adopt and adapt
  – Multiple pathways

■ Distinct CS1 sections feeding into a common CS2
  – CS1E for experienced students
  – Common assignments and grading
  – Enrichment without advantage

■ Outcomes
  – Student satisfaction with more and more divers emajors
  – Increase in diversity – a leader amongst peers
Lay of the land

- Vision – rose-tinted glasses
  - CS1
    - Students **without** experience
  - CS1E
    - Students with experience
- In practice
  - GPA-focused students picked CS1
A Learner-focused Approach
Outside the box

- Need a CS1X
  - Section for students without experience
  - Administrative-feedback – how about a section for women only
Outside the box

- Need a CS1X
  - Section for students without experience
CS1X founding principles

- Open to *anyone* without prior programming experience
- *Integrate* lecture and laboratory
- Use pedagogies effective for *all* types of student

- *Follow a scaffolded, active learning, stereotype threat-free, lab-centric approach, for achieving mastery of concepts and skills*
What’s it all about

- Change expectations
- Guided discovery
- Active collaborative learning
- Integrated lecture and lab
- Class culture of success
- Encouraging pedagogy
- Interesting examples
- Constant recruiting
What’s it all about

- **Change expectations**
- Guided discovery
- Active collaborative learning
- Integrated lecture and lab
- Class culture of success
- Encouraging pedagogy
- Interesting examples
- Constant recruiting
What’s it all about

■ *Change expectations*
■ Guided discovery
■ Active collaborative learning
■ Integrated lecture and lab
■ Class culture of success
■ Encouraging pedagogy
■ Interesting examples
■ Constant recruiting

Writing poetry
For my new computer class
This is an odd start

Difficult or fun
Will I be good at coding
I’m ready to try

My palms are sweaty
My hands shake from nervousness
Coding sounds intense
What’s it all about

- Change expectations
- Guided discovery
- Active collaborative learning
- Integrated lecture and lab
- Class culture of success
- Encouraging pedagogy
- Interesting examples
- Constant recruiting
What’s it all about

- Change expectations
- **Guided discovery**
- Active collaborative learning
- Integrated lecture and lab
- Class culture of success
- Encouraging pedagogy
- Interesting examples
- Constant recruiting

Software

- A major activity throughout the course will be to write Python programs. Before anything can happen you need to download and install two programs:
  - The downloads should be done prior to the first class meeting.
  - There are several variants of Python. We will be using Python 3.5.1. It is important to install Python 3.5.1 before class begins.
    - OS X: [download](#)
    - Windows: [download](#)
- When writing a Python program we will use an Integrated Development Environment (IDE) which will allow us to write, run, and debug Python programs.
  - OS X: [download](#)
  - Windows: [download](#)
What’s it all about

- Change expectations
- **Guided discovery**
- Active collaborative learning
- Integrated lecture and lab
- Class culture of success
- Encouraging pedagogy
- Interesting examples
- Constant recruiting
What’s it all about

- Change expectations
- Guided discovery
- **Active collaborative learning**
- Integrated lecture and lab
- Class culture of success
- Encouraging pedagogy
- Interesting examples
- Constant recruiting
What’s it all about

- Change expectations
- Guided discovery
- Active collaborative learning
- *Integrated lecture and lab*
- Class culture of success
- Encouraging pedagogy
- Interesting examples
- Constant recruiting
What’s it all about

- Change expectations
- Guided discovery
- Active collaborative learning
- Integrated lecture and lab
- Class culture of success
- Encouraging pedagogy
- Interesting examples
- Constant recruiting
What’s it all about

- Change expectations
- Guided discovery
- Active collaborative learning
- Integrated lecture and lab
- Class culture of success
- **Encouraging pedagogy**
- Interesting examples
- Constant recruiting
What’s it all about

- Change expectations
- Guided discovery
- Active collaborative learning
- Integrated lecture and lab
- Class culture of success
- Encouraging pedagogy
- *Interesting examples*
- Constant recruiting
What’s it all about

- Change expectations
- Guided discovery
- Active collaborative learning
- Integrated lecture and lab
- Class culture of success
- Encouraging pedagogy
- **Interesting examples**
- Constant recruiting
What’s it all about

- Change expectations
- Guided discovery
- Active collaborative learning
- Integrated lecture and lab
- Class culture of success
- Encouraging pedagogy
- *Interesting examples*
- Constant recruiting
What’s it all about

- Change expectations
- Guided discovery
- Active collaborative learning
- Integrated lecture and lab
- Class culture of success
- Encouraging pedagogy
- *Interesting examples*
- Constant recruiting
What’s it all about

- Change expectations
- Guided discovery
- Active collaborative learning
- Integrated lecture and lab
- Class culture of success
- Encouraging pedagogy
  - *Interesting examples*
- Constant recruiting
What’s it all about

- Change expectations
- Guided discovery
- Active collaborative learning
- Integrated lecture and lab
- Class culture of success
- Encouraging pedagogy
- *Interesting examples*
- Constant recruiting

---

**Computing Interests Survey**

Offering students interesting and meaningful examples that illustrate and apply computing concepts can improve learning. Your answers to the following optional and anonymous survey will help computer science instructors accomplish that goal.

Please rate your interest in the following topics as possible class and homework examples on a scale from extremely uninterested to extremely interested.

Information about a topic can be displayed by rolling the mouse over the topic.
What’s it all about

- Change expectations
- Guided discovery
- Active collaborative learning
- Integrated lecture and lab
- Class culture of success
- Encouraging pedagogy
- **Interesting examples**
- Constant recruiting

<table>
<thead>
<tr>
<th>Card games</th>
<th>Photo manipulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Photo manipulation</td>
<td>Medical diagnosis</td>
</tr>
<tr>
<td>Business applications</td>
<td>Card games</td>
</tr>
<tr>
<td>Music player</td>
<td>Language Translation</td>
</tr>
<tr>
<td>Password security</td>
<td>Music player</td>
</tr>
<tr>
<td>Medical diagnosis</td>
<td>Password security</td>
</tr>
<tr>
<td>Language translation</td>
<td>Personality typing</td>
</tr>
<tr>
<td>Training zone</td>
<td>Instant messaging</td>
</tr>
<tr>
<td>Connect four</td>
<td>Photo mosaics</td>
</tr>
</tbody>
</table>
Chrestomathics

\[ 5 \cdot \pi \cdot a \cdot b \cdot b / 24 \]

\[ \frac{d \cdot e}{5 \cdot \pi \cdot a \cdot b \cdot b / 24} \]
import math

LOADING_FACTOR = 0.698

length = input( 'Enter jelly bean length (cm): ' )
length = float( length )

diameter = input( 'Enter jelly bean diameter (cm): ' )
diameter = float( diameter )

jar_size = input( 'Enter jar size (mL): ' )
jar_size = float( jar_size )

bean_size = 5 * math.pi * length * diameter * diameter / 24

usable_volume = jar_size * LOADING_FACTOR

count = int( usable_volume / bean_size )

print( 'Beans: ', count )
Ubiquitous

- Change expectations
- Guided discovery
- Active collaborative learning
- Integrated lecture and lab
- Class culture of success
- Encouraging pedagogy
- Interesting examples

**Constant recruiting**
Resources

- CS 1X offerings
- CS Teaching Tips
- Chicago CSTA
- Nifty assignments
- Problems of interest

CS1X Archives

As of Fall 2015, Java was no longer the language of choice in our introductory programming courses. Python became the recommended language.

<table>
<thead>
<tr>
<th>CS1X offerings</th>
<th>Python-based</th>
<th>Java-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spring 2017</td>
<td>Spring 2015</td>
<td></td>
</tr>
<tr>
<td>Fall 2016</td>
<td>Fall 2014</td>
<td></td>
</tr>
<tr>
<td>Spring 2016</td>
<td>Spring 2014</td>
<td></td>
</tr>
<tr>
<td>Fall 2015</td>
<td>Fall 2013</td>
<td></td>
</tr>
<tr>
<td>Spring 2013</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall 2012</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spring 2013</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Resources

- CS 1X offerings
- CS Teaching Tips
- Chicago CSTA
- Nifty assignments
- Problems of interest

Computing Interests Survey

Offering students interesting and meaningful examples that illustrate and apply computing concepts can improve learning. Your answers to the following optional and anonymous survey will help computer science instructors accomplish that goal.

Please rate your interest in the following topics as possible class and homework examples on a scale from extremely uninterested to extremely interested.

Information about a topic can be displayed by rolling the mouse over the topic.
Success!

Best learning experience of my professional career

One of the most diverse and successful programs in the country
That’s all folks

- Community, baby steps, interesting examples, reinforcement, sociability

- Female-, URM-, all- people friendly

- Attracts all students

- Equal opportunity success
Thank you
Reflection

- Can your students tell you want to be leading the class? How?
- How do you make a positive environment?
- What should others copy from you?
- How has your class demographics changed since you started teaching computing? Why?
- How do you handle student show-offs?
Reflection

- Why do you think I favor students seeing me as a person is important? Would that also hold for you?
- What are your best pedagogic practices?
- Share your best assignments and examples.
- How do your favorites motivate your students?
Reflection

■ How often do you try to recruit students to follow-on computing classes?

■ What percentage of your students take follow-on computing classes?

■ What classroom and teaching practices are you going to change?

■ What classroom and teaching practices should you change, but won’t? How can you overcome that barrier?
Reflection

- Does your curriculum need updating?
- What do you think needs changing?
- What are the barriers for change?
- What do you think about having multiple pathways?
Reflection

■ What do you think about having multiple pathways?
■ Would multiple pathways be helpful at your institution? Why?
■ What are some other CS 1 variations can be tried?