Thanks for inviting me!

• Tell you about SIGCSE from my experiences

• Then – Engaging Students in Active Learning of CS Concepts
Professor of the Practice
Duke University
Computer Science Department

• What does that mean?
• Focus on Education in the discipline
• Assistant, Associate, Professor
• 20% of Arts and Science faculty
• Also use position in other ways
  • Hire journalist – Washington Post
  • Industry Liaison
Back in 1989

• PhD at Purdue University

• Assistant Professor at Rensselaer Polytechnic Institute

• First Task – combine first two computer science courses with automata theory
SIGCSE

• 1990 – interested in CS Education
• 1992 – my first SIGCSE conference
• 1996 – my first ITiCSE conference
• 2006 – my first ICER workshop
• 2015 – my first ACM India conference!
• 2015 – my first iSIGCSE event!
Never missed a SIGCSE Symposium since I started going

- 1997
- 2000

First son  Second son  Brought help
What SIGCSE has meant to me

• Great colleagues
  • Friendliest people
  • Friendliest conferences
• Learn about innovative pedagogy
  • Use in my courses
• Share
  • My ideas
  • My curriculum materials and software
Examples of Panels at SIGCSE Symposium

• 90’s – AP Computer Science transition from Pascal to C++

• Mid - 90’s – AP CS transition from C++ to Java

• What to teach in the first course

• Policies for getting computing into K-12

• ACM Curriculum
Special Session – Nifty Assignments
nifty.Stanford.edu

- Example: Evil Hangman
- First do a hangman assignment
  - Randomly select a word
  - Guess letters until spelled word
  - Only get so many tries

```python
# letters in word: 8
# guesses to hanging: 7
__ __ __ __ __ __
misses left: 7
guesses so far: e
  no e
__ __ __ __ __ __
misses left: 6
guesses so far: e
  guess letter: a
  no a
__ __ __ __ __ __
misses left: 5
guesses so far: a e
  guess letter: o
  no o
__ __ __ __ __ __
misses left: 4
guesses so far: a e o
  guess letter: u
  no u
__ __ __ __ __ __
misses left: 3
guesses so far: a u e o
  guess letter: i
  __ __ i __ i __
misses left: 3
```
Evil Hangman

• Make it difficult to win
  • Keep changing the word after the user guesses
  • After each guess, create the largest set of words that could be the match

• Word is: truck
• User guesses: u
• Change secret word to a word without ‘u’
<table>
<thead>
<tr>
<th>Secret Word</th>
<th>Possible Words</th>
<th>Progress</th>
<th>Missed Letters</th>
<th>Guessed Letters</th>
</tr>
</thead>
<tbody>
<tr>
<td>jumpy</td>
<td>87</td>
<td>_ _ _ _ _</td>
<td>a i s e o</td>
<td>u</td>
</tr>
<tr>
<td>bulgy</td>
<td>47</td>
<td>_ u _ _ _</td>
<td>a i s e o</td>
<td>b</td>
</tr>
<tr>
<td>lucky</td>
<td>33</td>
<td>_ u _ _ _</td>
<td>a b e i o s</td>
<td>c</td>
</tr>
<tr>
<td>funny</td>
<td>21</td>
<td>_ u _ _ _</td>
<td>a c b e i o s</td>
<td></td>
</tr>
</tbody>
</table>
Check out other Nifty Assignments
Hope your SIGCSE grows and grows

• Expect to see an annual conference in India...

• Expect impact in both undergraduate curriculum and schools
Outline

- Lecture formats
- Prepare for Class
- Class participation
- Engaging with Software
- Peer-led team learning
- Engaging in other ways
Motivation

• Traditional way of teaching
  • Professor Lectures
  • What students hear about 87%

 BLAH BLAH BLAH BLAH BLAH BLAH
Active Learning - Workshop Format
“Flipped” Classroom

- Lecture for 10-20 minutes
- Students work on problem with computers in pairs
- Bring students back together
- Pair programming
Alternative

• Work in pairs
• Everyone has their own laptop
Groups/Pairs

• Assigned

CompSci 4 Section 1
Pairs as of October 22, 2009

Front of room

G1 G2  G3 G4  
G5  G6  G7  G8  G9
G10 G11 G12 G13 G14 G15
G16 G17 G18 G19 G20
G21 G22 G23 G24 G25

Group 1
Starkweather, Clara cks15@duke.edu
Dinkins, Tiffany tiffany.dinkins@duke.edu

Group 2
Listenbee, Kamerria kkl7@duke.edu
Brown, Dwayne dcb26@duke.edu

Group 3


Outline

• Lecture formats
• Prepare for Class
• Class participation
• Engaging with Software
• Peer-led Team Learning
• Engaging in other ways
Read the book

- Read before coming to class
  - Ready to work in class
- Reality
  - Run out of time to read, not prepared
- Bring on – Reading quizzes
  - Online
  - Turn off when class starts

Question 4 of 8
What is the output of the following:
```
alist = [6, 3, 4, 9]
del alist[1]
print alist
```
- A. [8]
- B. [3, 4, 9]
- C. [6, 3, 4]
- D. [8, 4, 9]
Have an engaging book....

• Runescape (Brad Miller)

Here is the program in activecode. Note that the function definition is the same as it was before. All that has changed is the details of how the squaring is done. This is a great example of “black box” design. We can change out the details inside of the box and still use the function exactly as we did before.

```
def square(x):
    runningtotal = 0
    for counter in range(x):
        runningtotal = runningtotal + x
    return runningtotal

toSquare = 10
squareResult = square(toSquare)
print("The result of", toSquare, "squared is", squareResult)
```
Electronic Textbooks (ebooks) engage students

- OpenDSA (Shaffer, Virgina Tech)
  - Algorithm animations built in
- runestoneinteractive.org (Brad Miller,
  - Several books (Python)
    - Python - try and run code built in
    - Quizzes
- Zyante.com – interactive textbooks
- Track student progress
- Requirements and design strategies for open source interactive computer science eBooks
  - ITiCSE 2013 Working Group (Korhonen, Naps, et al)
Use engaging and visual tools
Python Tutor

```python
1  scores = [10, 8, 3, 9]
2  list2 = scores
3  list3 = scores[:]
4  scores[2]=5
```
Problem Solving with Feedback

- APT – Algorithmic Problem Tester
- Test one function
- Runs on multiple inputs

# of correct: 14 out of 17
Outline

• Lecture formats
• Prepare for Class
• Class participation
• Engaging with Software
• Engaging in other ways
Ways to Select students to answer questions

• Problem – same students always eager
• How do you get other students to participate?
  • Randomly call on them
  • Pick A Student program
  • Work in groups – call on group
  • Assigned groups – call on group numbers
Randomly Select a Student
Pick A Student Program

Collect pictures of students

program that cycles through and randomly picks one

Remove, then start again

From Owen Astrachan
Mystery While

* Required

**NETID of person 1** *
Example: abc123

**NETID of person 2**
Example: abc123

**NETID of person 3**
Example: abc123

**NETID of person 4**
Example: abc123

**Names of people filling out form** *
(first and last name for each person, separate each name by a comma)
What does Mystery2 do (in words)?*

What does Mystery3 do?*
(in words)
<table>
<thead>
<tr>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What does Mystery3 do?</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>It counts every character in a word except for lowercase &quot;e&quot;s</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>It counts the number of characters in the word that aren't lowercase e's</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>It is counting the number of characters in the word that are not e's</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Count the number of characters that are not 'e' in the word</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mystery 3 returns a given a given word without the lowercase e's.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Counts the lowercase es in the word.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>It returns the number of characters in a word that are not e.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Counts all of the letters in word that aren't 'e'</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Counts the number of characters that are not e in the word.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Counts all the characters that aren't e</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>counts how many letters there are that are not &quot;e&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mystery 3 counts the characters in a string that are not 'e', then returns the total count.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Setting up Google Forms

• Make it easy for students to get form

![Bitly interface showing custom bitlink creation](image)
Outline

• Lecture formats
• Prepare for Class
• Class participation
• Engaging with Software
  • JFLAP
  • ALICE
• Engaging in other ways
Yesterday – talked about JFLAP

- Java Formal Languages and Automata Package
- Instructional tool to learn concepts of Formal Languages and Automata Theory
- Topics:
  - Regular Languages
  - Context-Free Languages
  - Recursively Enumerable Languages
  - Lsystems
- With JFLAP your creations come to life!
JFLAP in CPS 140/334 at Duke

• Lecture
  • Demo how to use JFLAP
  • Solve a problem using JFLAP with the class
    • Give them time to think about it first
    • Either from scratch, or give them an attempt that needs to be fixed
  • Teach conversions using an example
  • Run examples, some with exponential time

www.cs.duke.edu/courses/spring14/compsci334
JFLAP in CPS 140 Homework

• Spring 2005 - 10 of 11 hwks used JFLAP
• Use JFLAP to check your answer
• Turn in a JFLAP file
• Grading – use multiple run window
  • Type in data once, it stays around
Use of JFLAP by Instructor

Showing how to layout items

Poor:

Better:
Use of JFLAP by Instructor

Is this correct for $a^n b^n c^n$?

How do we fix it?
Use of JFLAP by Instructor

Experimenting with Difficult Concepts

Nondeterminism: $ww^R$

• Students attempt at desk - difficult: want to find the “middle”
• Instructor solves with class using JFLAP
Use of JFLAP by Instructor

Testing Student Programs

Diagram:

```
q0: a; a, R | □; a, R
    b; b, S | □; □, L
q2: c; c, R | b; c, R
    □; □, R | □; □, R
    □; □, R | □; □, R
q1: b; b, R | a; b, L
q3: □; □, R | □; □, R
```

Table:

<table>
<thead>
<tr>
<th>Input 1</th>
<th>Input 2</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>aabbcc</td>
<td></td>
<td>Accept</td>
</tr>
<tr>
<td>aaabbbccc</td>
<td></td>
<td>Accept</td>
</tr>
<tr>
<td>bc</td>
<td></td>
<td>Reject</td>
</tr>
<tr>
<td>aabbc</td>
<td></td>
<td>Reject</td>
</tr>
<tr>
<td>aabcc</td>
<td></td>
<td>Reject</td>
</tr>
<tr>
<td>abbcc</td>
<td></td>
<td>Reject</td>
</tr>
<tr>
<td>abc</td>
<td></td>
<td>Accept</td>
</tr>
<tr>
<td>aabbc</td>
<td></td>
<td>Reject</td>
</tr>
<tr>
<td>aabbc</td>
<td></td>
<td>Reject</td>
</tr>
</tbody>
</table>

Run Inputs Clear Enter Lambda
Use of JFLAP by Instructor

Relate to other CS Concepts

Running Time
• Consider $a^n b^n c^n$
  • one-tape TM $O(n^2)$
  • two-tape TM $O(n)$
Other Uses of JFLAP by Instructor

• Demonstrate Nondeterminism
• Demonstrate the running of a CFG to a PDA using LR method

Which lookahead do you choose?

• Demonstrate a transformation from one form to another

Example: PDA to CFG

• And many other uses...
JFLAP Student Use

• Recreate and experiment with instructor’s examples
• Use with Homework
• A study aid - create additional examples
  • explore concepts in depth
  • weaker students get more feedback
Alice Programming Language

• Create interactive stories or games
• Learn programming in an easy way, drag-and-drop your code
• Problem solving with visual feedback
  • Logical thinking, Computational thinking
• Along the way, learn computer science concepts:
  • Loops, classes, methods, functions, arrays
Alice Developed by Randy Pausch

- Carnegie Mellon University
- Virtual Reality Researcher
- Wrote the Last Lecture
- Died of Pancreatic Cancer in 2008
More on ... Alice Programming Language

• Has libraries of 3D objects

• Keeps Track of objects you select
Objects Have Multiple Parts that are moveable
Object Position

• Objects
  • Are positioned in 3D space
  • Have six degrees of freedom
Alice Code is Easy to Learn

Select Code, Drag-and-Drop code in program
Play Alice Animation

• Chicken rises, cow turns head and talks
Adventures in Alice Programming
Grades 5-12 Outreach

www.cs.duke.edu/csed/alice/aliceInSchools
Adventures in Alice Programming

• 2-week Teacher workshops
  • Over 200 teachers, middle school, high school, some elementary
  • First week Teach Alice, Practice
  • Second week - Develop Lesson Plans
• One-week follow-up workshop the following summer
  • Summers 2008-2015, funding for lodging

• Main Sites:
  • Duke University, Durham, NC
  • Charleston/Columbia, SC
  • San Jose, CA (starting 2014)
Getting Started Tutorials

• One-hour tutorial
  • Covers placing objects, setting camera views, basic commands, writing methods and events

• 3 versions of it – pick story your students will like
Getting started tutorial
One Hour

Hey! Welcome to my island!
Science Example
How a volcano is formed
Science – Population Change
now we'll graph the data in a bar chart to see how the population changed over time.
Foreign Language simple

¡Bienvenido al programa de cocinar!
Focus on math
Math Example – Plotting Numbers

I am going on a bike ride
Math Story on Fractions
Curriculum Materials

• Over 90 tutorials available for free
• Beginner, advanced, challenges, projects
• Paper handouts and video
• Teacher lesson plans in many disciplines
Tutorial for Project: Book Report

Charlotte's Web
by E.B. White
Tutorial for Adventure Game – Find objects in order
Sample tutorial: Scene Change
New Tutorial – Camera views following a person
Harry Potter Challenge

• Mix of programming and math challenges

Hailey Programmer and the Goblet of Java

You will receive a password at the end of each level that will be used to unlock the next level. WRITE THESE DOWN!
If this is your first time playing, select Charms.
Harry Potter – Math/computing
Level 1 Charms - before
Harry Potter – Math/Computing
Level 1 Charms - after
Other Example Challenges

Boat

Calculator

To win this game, you must steer the boat through each ring and beat the clock. You receive one point for each ring, and there are 10 rings, so if your score is less than 10 at the end, you lose!
Enhanced Calculator!
Fun with Alice

The ITiCSE 2014 boat trip
What a 6th grader can do with Alice
- teacher Chari Distler
No Superheros in Alice
Outline

• Lecture formats
• Prepare for Class
• Class participation
• Engaging with Software
• Peer-led team learning
• Engaging in other ways
Motivation for Peer-Led Team Learning

• Goals
  • Increase number of women and underrepresented groups
  • Increase retention and enthusiasm

• Approaches
  • Active Recruiting of Incoming First-year students
  • Optional/Required of registered students
8 NSF-FUNDED SCHOOLS
What is PLTL?

• Related to a course
  – Students solve problems in small groups (4-8 students) weekly in addition to regular class meeting
  – Interesting exercises to be solved as a group
  – Led by trained undergraduate student leaders who facilitate group learning

• Used in Chemistry for about 12 years, www.pltl.org

• Beneficial to both students and student leaders
Groups != Discussion Section

• Students helping, learning from other students
• Less authoritative; liberate and empower students
• Promote Active Learning, encourage teamwork
• More fun!
Why PLTL?

• Factors affecting intellectual development in college
  • Student/faculty interaction outside the classroom
  • Involvement on campus through various forms of community-building activities
  • Involvement with student peer groups
  • “peer group – the most potent source of influence on growth and development during the undergraduate years.”
Cone of Learning (Edgar Dale)

After 2 weeks we tend to remember...

- 10% of what we read
- 20% of what we hear
- 30% of what we see
- 50% of what we hear and see
- 70% of what we say
- 90% of what we say and do

Nature of involvement:

- Lecture
- PLTL

Effects on Students

• Better/deeper understanding of material
• Lower drop rates
• Better grades (usually)
• Formation of social groups
• Very high satisfaction
Effects on Peer Leaders

• Better understanding of the material
• Increased confidence to continue in CS
• Appreciation for different teaching/learning styles
• Improved leadership skills
• Collegial relationship with faculty
What is ESP?

• Emerging Scholars Program
  – Used in math and science courses
  – Recruits under-represented groups
  – Works in small groups on challenging problems

• Benefits
  – Earn Higher Grades
  – Increases enthusiasm for math and science

Defining PLTL in CS (also called ESP-PLTL)

• Small groups meet related to a course
  • Not everyone from the course
  • Build friendships to help support you through major

• Active recruiting

• Aim for gender balance

• Undergraduate peer leaders

• Solve challenging problems
Peer-Led Team Learning in CS (PLTL in CS)

• Combines components from PLTL and ESP

• Eight Universities – Fall 2005 – Spring

  Beloit College (WI)  Purdue University (IN)
  Duke University (NC)  Rutgers University (NJ)
  Georgia Tech (GA)  University of Wisconsin Madison (WI)
  Loyola College (MD)  University of Wisconsin Milwaukee (WI)

• www.pltlcs.org

Supported by the National Science Foundation collaborative Grants CNS-0420436, 0420343, 0419340, 0420433, 0420358, 0420312, 0420368, 0420337, 0638510 and 0638499 and a donation from Microsoft.
PLTL in CS variations among 8 universities

• Some focus on non-majors course
• Some focus on CS 1
• Some focus on both (one year)
• Some have just women, most are mixed
• Some include everyone, most are subset

• All use active recruiting and undergraduate peer leaders
• All use problem solving in small groups outside of main class period
Duke University - “PLTL in CS” version
Emerging Scholars Program (DES)

- One year program – four courses total
  - First semester
    - Main course: Non-majors course: CPS 4 (Alice) (1 credit)
    - Problem Solving Seminar course: CPS 18S (1/2 credit)
  - Second Semester
    - Main Course: CS 1 course: CPS 6 (Java)
    - Problem Solving Seminar course: CPS 18S (1/2 credit)
  - Active Recruiting (email to targetted groups, accepted student fairs, invite students in main course)
  - Gender balanced
  - Outside Speaker/Field Trip
  - Undergraduate Peer Leaders in Problem Solving Seminar
CompSci 18S: Problem Solving Seminar

• 2 peer leaders, about 12 students, (1 professor)
• Solve problems in groups of 4
• Either general computer science problems or related to the main course
• Subset of students from main course, those who want the group experience
• Peer leaders trained in workshop, meet weekly
2 Main Courses: Non-majors (Alice) and CS 1 (Java)

• Workshop format
  – Lecture 10-20 minutes
  – Students program rest of class
  – Students work in pairs ("pair programming")
    • Two people, two laptops, consult a lot
  – Assigned seats and pairs, change every 2-3 weeks

• About 35-50 students
2 Main Courses: Undergraduate role

• About 8-10 undergraduate teaching assistants

• Roles:
  • Attend the “workshop lecture” to assist
  • Meet weekly
  • Grade and hold consulting hours
  • Includes the two peer leaders from the problem solving seminar
Example of Problem Solving: Be A Robot

• Group of 4 – brain, eyes, 2 hands
• Only brain knows what you are building
• Only eyes can see
• Must work together precisely like a robot
Example of Problem Solving: Sorting

• Give students an envelope of words – each word on a separate sheet of paper
• In groups, sort the words
• Write down the algorithm you designed
Sorting (cont)
Questions to think about when organizing the words...

• How many words are there?
• How many words start with “m”?  
• How many words end in “k”?  
• What is the 66\(^\text{th}\) word?  
• How many words have “her” in them?  
• What is your favorite word?
Example of Problem Solving - Graphs

• Graph of all friends (of everyone in class, at your university)

• Problems
  • Find number of friends of friends of someone
  • Find the center of the graph – person with smallest sum of shortest distances
Other Examples

- Finite State Machines
- Turing Machines
- Random Numbers
- Compression (Huffman coding)
- Sudoku, Jumble
- L-Systems
- Genomics
Results from Study

• Susan Horwitz, Susan Rodger, Maureen Biggers, David Binkley, C. Kolin Frantz, Dawn Gundermann, Susanne Hambrusch, Steven Huss-Lederman, Ethan Munson, Barbara Ryder, and Monica Sweat, Using Peer-Led Team Learning to Increase Participation and Success of Under-Represented Groups in Introductory Computer Science, Fourtieth SIGCSE Technical Symposium on Computer Science Education, 2009
Results:
Why did women enroll in PLTL in CS?

41 women responded in 2005-06

60.5% mailed invitation
15.6% other
12.8% info during orientation
9.8% academic advisor recommendation
9.8% class announcement
4.9% parent recommendation
## Results - Why enroll in main course?

31 female/49 male responses 2005 (select all that apply)

<table>
<thead>
<tr>
<th></th>
<th>M</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>71.0%</td>
<td>I received an invitation</td>
</tr>
<tr>
<td>M</td>
<td>22.5%</td>
<td>To see whether I enjoy CS</td>
</tr>
<tr>
<td>F</td>
<td>67.7%</td>
<td>Meets requirement for my major</td>
</tr>
<tr>
<td>M</td>
<td>28.6%</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>29.0%</td>
<td>I know I am interested in CS</td>
</tr>
<tr>
<td>M</td>
<td>40.8%</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>25.8%</td>
<td>Programming is useful job-market skill</td>
</tr>
<tr>
<td>M</td>
<td>79.6%</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>19.4%</td>
<td>I plan to major in CS</td>
</tr>
<tr>
<td>M</td>
<td>18.4%</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>16.1%</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>57.1%</td>
<td></td>
</tr>
</tbody>
</table>
Results - Recruiting

- Percentage of women and minorities was higher in ESP-PLTL
- This is overall institutions from 2005-2007.

<table>
<thead>
<tr>
<th></th>
<th>ESP-PLTL</th>
<th>Main Course</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#</td>
<td>%</td>
</tr>
<tr>
<td>Female</td>
<td>122</td>
<td>33.4%</td>
</tr>
<tr>
<td>Minority</td>
<td>43</td>
<td>11.8%</td>
</tr>
</tbody>
</table>
## Retention Data

<table>
<thead>
<tr>
<th></th>
<th>ESP-PLTL</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#</td>
<td>%</td>
<td>#</td>
<td>%</td>
<td>#</td>
<td>%</td>
</tr>
<tr>
<td>Completed</td>
<td>383</td>
<td>93.2%</td>
<td>2363</td>
<td>88.0%</td>
<td>2746</td>
<td>88.7%</td>
</tr>
<tr>
<td>Dropped</td>
<td>28</td>
<td>6.8%</td>
<td>323</td>
<td>12.0%</td>
<td>351</td>
<td>11.3%</td>
</tr>
<tr>
<td>Total</td>
<td>411</td>
<td>100.0%</td>
<td>2686</td>
<td>100.0%</td>
<td>3097</td>
<td>100.0%</td>
</tr>
</tbody>
</table>
Final Grade Data, all Institutions 2005-2007

<table>
<thead>
<tr>
<th></th>
<th>All ESP-PLTL</th>
<th>All Non-ESP-PLTL</th>
<th>Total (All Students)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#</td>
<td>%</td>
<td>#</td>
</tr>
<tr>
<td>B or better</td>
<td>219</td>
<td>80.2%</td>
<td>1130</td>
</tr>
<tr>
<td>Less than B</td>
<td>54</td>
<td>19.8%</td>
<td>522</td>
</tr>
<tr>
<td>Total</td>
<td>273</td>
<td>100.0%</td>
<td>1652</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>ESP-PLTL Female</th>
<th>Non-ESP-PLTL Female</th>
<th>Total (All Females)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#</td>
<td>%</td>
<td>#</td>
</tr>
<tr>
<td>B or better</td>
<td>70</td>
<td>83.3%</td>
<td>295</td>
</tr>
<tr>
<td>Less than B</td>
<td>14</td>
<td>16.7%</td>
<td>126</td>
</tr>
<tr>
<td>Total</td>
<td>84</td>
<td>100.0%</td>
<td>421</td>
</tr>
</tbody>
</table>
Advantages for Peer Leaders (telephone interview)

• Common themes emerged
  • Improved Leadership skills
  • Opportunity to try out educator role
  • Reinforcement of understanding CS concepts
  • Increased confidence to continue in field
  • Friendships with students
  • Would recommend experience to others
Summarizing results

• Active Recruiting increased number of women
  • Email/mailed invitation was most effective
• Retention of PLTL in CS students was higher
• Grades of PLTL in CS students was higher
• Friendships and Bonding occurred with students
• Advantages for Peer Leaders too
• PLTL in CS workshop April 2007 at Duke
Outline

• Lecture formats
• Prepare for Class
• Class participation
• Engaging with Software
• Peer led team learning
• Engaging in other ways
Engaging students in a group activities/large course

• Problem Solving in groups
  • Clickers, Google forms – compare results
  • Flip Classroom, reading quizzes (turn off at start of class)

• Acting out stories, games
  • *Everything I needed to know about teaching...* - Pollard, Duvall (SIGCSE 2007)

• Acting out algorithms with the whole class
  • Make a binary tree with the whole class
  • Calculate the height of the tree
  • *Making Lemonade ... large lecture classes* – Wolfman (SIGCSE 2002)

• Acting out algorithms with a subset of students
  • Sorting algorithms – selection sort, insertionsort, etc
  • CS Unplugged activities
Example of Computer Science concept

- Children’s book
This was no time for play.
This was no time for fun.
This was no time for games.
There was work to be done.
All that deep,
Deep, deep snow,
All that snow had to go.

When our mother went
Down to the town for the day,
She said, “Somebody has to
Clean all this away.
Somebody, SOMEBODY
Has to, you see.”
Then she picked out two Somebodies.
Sally and me.
Well...
There we were.
We were working like that
And then who should come up
But the CAT IN THE HAT!

“Oh-oh!” Sally said.
“Don’t you talk to that cat.
That cat is a bad one,
That Cat in the Hat.
He plays lots of bad tricks.
Don’t you let him come near.
You know what he did
The last time he was here.”
"Play tricks?" laughed the cat.
"Oh, my my! No, no, no!
I just want to go in
To get out of the snow.
Keep your mind on your work.
You just stay there, you two.
I will go in the house
And find something to do."

Then that cat went right in!
He was up to no good!
So I ran in after
As fast as I could!
Do you know where I found him?
You know where he was?
He was eating a cake in the tub!
Yes he was!
The hot water was on
And the cold water, too.
And I said to the cat,
“What a bad thing to do!”

“But I like to eat cake
In a tub,” laughed the cat.
“You should try it some time,”
Laughed the cat as he sat.
The water ran out.
And then I saw the ring!
A ring in the tub!
And, oh boy! What a thing!
A big long pink cat ring!
It looked like pink ink!
And I said, “Will this ever
Come off? I don’t think!”
“Have no fear of that ring,”
Laughed the Cat in the Hat.
“Why, I can take cat rings
Off tubs. Just like that!”

Do you know how he did it?
WITH MOTHER’S WHITE DRESS!
Now the tub was all clean,
But her dress was a mess!
But the cat just stood still.
He just looked at the bed.
"This is not the right kind of a bed,"
The cat said.
"To take spots off this bed
Will be hard," said the cat.
"I can't do it alone,"
Said the Cat in the Hat.
"It is good I have some one
To help me," he said.
"Right here in my hat
On the top of my head!
It is good that I have him
Here with me today.
He helps me a lot.
This is Little Cat A."
And then Little Cat A
Took the hat off his head.
"It is good I have some one
To help me," he said.
"This is Little Cat B.
And I keep him about,
And when I need help
Then I let him come out."
And then B said,
"I think we need Little Cat C.
That spot is too much
For the A cat and me.
But now, have no fear!
We will clean it away!
The three of us! Little Cats B, C and A!"
“Come on! Take it away!”
Yelled Little Cat A.

“I will hit that old spot
With this broom! Do you see?
It comes off the old bed!
It goes on the T.V.”

And then Little Cat B
Cleaned up the T.V.

He cleaned it with milk,
Put the spot in a pan!
And then C blew it out
Of the house with a fan!
“But look where it went!”
I said. “Look where it blew!
You blew the mess
Out of the house. That is true.
But now you made Snow Spots!
You can’t let THEM stay!”

“Let us think about that now,”
Said C, B and A.
"With some help, we can do it!"
Said Little Cat C.
Then POP! On his head
We saw Little Cat D!
Then, POP! POP! POP!
Little Cats E, F and G!

"We will clean up that snow
If it takes us all day!
If it takes us all night,
We will clean it away!"
Said Little Cats G, F, E, D, C, B, A.
Then Little Cat G
Took the hat off his head.
"I have Little Cat H
Here to help us," he said.

"Little Cats H, I, J,
K, L and M.
But our work is so hard
We must have more than t!
We need Little Cat N.
We need O. We need P.
We need Little Cats Q, R, S
U and V."
"Look close! In my hand
I have Little Cat V.
On his head are Cats W,
X, Y and Z."

"Z is too small to see.
So don’t try. You can not.
But Z is the cat
Who will clean up that spot!"
“Now here is the Z
You can’t see,” said the Cat.
“And I bet you can’t guess
What he has in his hat!

“He has something called voom.
Voom is so hard to get,
You never saw anything
Like it, I bet.
Why, Voom cleans up anything
Clean as can be!”
Then he yelled,
“Take your hat off now,
Little Cat Z!
Take the Voom off your head!
Make it clean up the snow!
Hurry! You Little Cat!
One! Two! Three! Go!”
Then the Voom . . .
It went voom!
And, oh boy! What a voom

Now, don’t ask me what
I never will know.
But, boy! Let me tell you
It DOES clean up snow!
“So you see!” laughed the Cat,
“Now your snow is all white!
Now your work is all done!
Now your house is all right!
And you know where my little cats a
Said the cat.
“That Voom blew my little cats
Back in my hat.
And so, if you ever
Have spots, now and then,
I will be very happy
To come here again . . . 
"... with Little Cats A, B, C, D...
E, F, G...
H, I, J, K...
L, M, N...
and O, P...
... and Q, R, S, T...
and Cat U and Cat V...
and Little Cats W
X
Y
and Z!"
Conclusions

• We have shown several ways to engage students with CS concepts and make those concepts come alive

• Questions?

• www.cs.duke.edu/~rodger