Through Visualization and Interaction, Computer Science Concepts Come Alive

Susan H. Rodger
Duke University
May 11, 2016
Evergreen State College

About Me
• Professor of the Practice of Computer Science
• Area: Visualization and Animation, Computer Science Education
• Passionate about education/diversity
  • SIGCSE Chair
  • ACM Education Policy Committee
  • CRA-W Board Member

About Me - Hobby – Baking Shape cakes

How do you make those cakes?
CS 1
Sorting Cookies

Outline

• Introduction
• CS Concepts Come Alive
  • Alice Programming Language
  • Algorithm Visualization
  • Automata Theory with JFLAP
  • Additional Ways to Engage with CS
• Diversity Efforts

Graduate School

• PhD Purdue University 1989
  • Computational Geometry
  • Parallel Scheduling Algorithms

CS Concepts Coming Alive

• What data structure is this?
YARN, in the shape of a binary tree. Subtrees made with a molecule kit. What is it?

2D-range tree

- Search in x-y plane
- Main tree organized by x-values
- Subtree organized by y values

Binary Search tree of points in the plane – sorted by X-value

Each subtree organized by y-value

Search each subtree by y-value

Different Types of Learners

- Learning Styles
  - Visual Learners
    - Learn through seeing
    - Learn best from visual displays
  - Auditory Learners
    - Learn through listening
    - Learn best through verbal lectures, discussions
  - Kinesthetic Learners
    - Learn through moving, doing and touching
    - Learn best through hands-on approach
How do you reach all three types?

• You must do all three!
  – Provide pictures, diagrams
  – Discuss what you are doing
  – Provide activities for trying it

Learner Engagement Taxonomy with visualization software

• Different forms of Learner engagement
  – No Viewing
  – Viewing
  – Responding
  – Changing
  – Constructing
  – Presenting

• ITiCSE Working Group Report 2002 (Naps et al.)

Outline

• Introduction
• CS Concepts Come Alive
  – Alice Programming Language
  – Algorithm Visualization
  – Automata Theory with JFLAP
  – Additional Ways to Engage with CS
• Diversity Efforts

Alice Programming Language

• Create interactive stories or games
• Learn programming in an easy way, drag-and-drop your code
• Problem solving with visual feedback
  • Objects are visual!
• Alice is free: www.alice.org
• Developed by Randy Pausch
More on ... Alice Programming Language

• Has libraries of 3D objects

• Keeps Track of objects you select

Objects Have Multiple Parts that are moveable

Alice Code is Easy to Learn

Select Code, Drag-and-Drop code in program

Play Alice Animation

• Chicken rises, cow turns head and talks
Why Alice?
• Lots of other great tools for teaching programming

Greenfoot

• Alice is easy to use, drag-and-drop, objects already exist
• Storytelling - Attractive to both girls and boys

Success - Alice attracts diverse group
• At Duke
  • CompSci 4 Spring 2005
    • 22 preregister, 30 enroll (12 female + 3 African Amer.)
  • CompSci 4 Fall 2005
    • 20 preregister, 31 enroll (17 female + 1 African Amer.)
  • CompSci 4 Fall 2006 – 2 sections
    • 64 students, 33 female, 7 African Amer.
  • CompSci 4 Fall 2007 – 2 sections
    • 84 students - > 50% female
  • CompSci 4 Fall 2008 – 2 sections
    • 100 students - > 50% female
  • Same for Spring 2009, Fall 2009...
  • Advertised in school paper
    • picture of ice skater
    • Web site of animations
  • This course is now CompSci 94

Computer Science Concepts come alive with Alice - Examples
• Objects are visible
• Variables
• Inheritance
• Lists
• Array

Example: Objects are visible
Getting Started Tutorial teaches
• Placing objects
• Moving objects
• Setting up Camera tripods and moving between views
• Using built in methods and writing your own
  • Dragon flapWings
• Gluing objects together
• Adding sound, 2D pictures to enhance world
Getting Started Tutorial – 3 part

Variables – Timer and Score

Variables – Scores/Timers

Game: Eragon

Example - Inheritance

- Start with a chicken object
- Rename it to TalentedChicken
- Change its color
- Resize it larger
- Add new methods (jump, fly, scurry)
- Add events for this chicken
- Save this new class TalentedChicken that inherits from the Chicken class

4 tasks to win the game
Example list

Example – Arrays
Shuffle, then Selection Sort

Outline

• Introduction
• CS Concepts Come Alive
  • Alice Programming Language
  • Algorithm Visualization
  • Automata Theory with JFLAP
  • Additional Ways to Engage with CS
• Diversity Efforts

Algorithm Visualization/Animation
Software/Aps/Videos

• Tango, Xtango, Samba, JSamba - Stasko (Georgia Tech)
• AnimalScript – Roessling (Darmstadt Univ of Tech, SIGCSE 2001)
• JHAVE – Naps (U. Wisc. Oshkosh, SIGCSE 2000)
• TRAKLA2 – Software Visualization Group – TKK Finland
• Lots of animations and systems on the web!
• Lots of videos of algorithm animations on the web!
Use of Algorithm Animation in CS 1/2

• Instructor
  • Make/Use animations for lecture
  • Stop/Pause – ask what will happen next
  • must be interactive

• Student
  • Create animations
  • Replay animations from lecture with same or new inputs

Lots of other software/programs for algorithm animation

• Red Black Tree – animation on [web page](http://aleph0.clarku.edu/~achou/cs102/examples/bst_animation/RedBlackTree-Example.html)
  
  ![Red Black Tree Animation](http://aleph0.clarku.edu/~achou/cs102/examples/bst_animation/RedBlackTree-Example.html)

  Student must have graduated. Link no longer works!

Another red-black tree animation

![Another red-black tree animation](http://www.ece.uc.edu/~franco/C321/html/RedBlack/redblack.html)

[TRAKLA2](http://www.cse.hut.fi/en/research/SVG/TRAKLA2/video/)
Python Tutor
Compute reverse of a list

1. `def reverse(numbers):`
2. `answer = []`
3. `for num in numbers:`
4. `answer.insert(0, num)`
5. `return answer`
6. `myList = [4, 7, 8, 3]`
7. `reversed = reverse(myList)`

Empty list
Compute reverse of a list

```python
def reverse(numbers):
    answer = []
    for num in numbers:
        answer.insert(0, num)
    return answer

myList = [4, 7, 8, 3]
reversed = reverse(myList)
```
**AlgoViz – Repository of Algorithm Visualizations**

- [AlgoViz.org](http://AlgoViz.org)

**Browse the Catalog**
- Linear Structures
- Search Structures
- Spatial Search Structures
- Misc. Data Structures
- Quadratic Sorts
- N log N sorts
- Misc. Sorts
- Search Algorithms
- Compression Algorithms
- Computational Geometry
- Graph Algorithms
- NP Completeness
- Numerical Algorithm
- Systems and Languages
- Algorithmic Techniques
- Misc. Topics

---

**Electronic Textbooks (ebooks) engage students**

- OpenDSA (Shaffer, Virginia 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)

---

**How to Think Like a Computer Scientist**

*Learning with Python: Interactive Edition 2.0*

**Index Operator: Working with the Characters of a String**

The indexing operator (Python uses square brackets to enclose the index) selects a single character from a string. The characters are accessed by their position or index value. For example, in the string shown below, the 14 characters are indexed left to right from position 0 to position 13.

```python
school = "Luther College"
```

Run and edit code in the book
Integrates in Python Tutor

Questions for feedback

Outline

• Introduction
• CS Concepts Come Alive
  • Alice Programming Language
  • Algorithm Visualization
  • Automata Theory with JFLAP
• Additional Ways to Engage with CS
• Diversity Efforts

Java Program to count number of words in a phrase

```java
public static void main() {
    String phrase = "baby fish should be fed three times a day";
    int numWords = 0;
    int pos = phrase.indexOf(" ");
    while (pos > 0) {
        numWords += 1;
        phrase = phrase.substring(pos + 1);
        pos = phrase.indexOf(" ");
    }
    numWords += 1;
    System.out.println("Number of words in phrase is " + numWords);
}
```

• Is this program syntactically correct?
First identify each word/token

```
public static void main() {

TOKEN   TYPE
public  keyword
static  keyword
void    keyword
main    variable
(       leftparen
)      rightparen
{       leftbrace

```

First identify each word/token (cont)

```
String phrase = "baby fish should be fed three times a day";

TOKEN   TYPE
String   keyword
phrase   variable
equals  equals
doublequote  string
doublequote  doublequote
;         semicolon

```

First identify each word/token (cont)

```
int numWords = 0;

TOKEN   TYPE
int     keyword
numWords variable
equals equals
0       int
;       semicolon

```

Define Rules for a valid program (Grammar)

```
<program> ::= <procDef> ( ) { <body> }
<program> ::= <procDef> (<arglist> ) { <body> }
<procDef> ::= public static void main
<procDef> ::= <pubtype> <returntype> <variable>
<body>   ::= <decllist> <stmtlist>
<decllist>::= <decl> ; <decllist>
<decllist>::= <decl> ;
<stmtlist> ::= <stmt> ; <stmtlist>
<stmtlist> ::= <stmt> ;

```

"baby fish ... day"

Define Rules for a valid program (Grammar)

<decl> ::= int <variable> = <integer> ;
<decl> ::= String <variable> = “<string> “ ;
<stmt> ::= while ( <cond> ) { <stmtlist> }
<stmt> ::= <variable> += <integer> ;
<stmt> ::= <variable> = <integer> ;
etc

Now derive the program using the rules (grammar)

<program> ::= <procDef> ( ) { <body> }

:= public static void main ( ) { <body> }

:= public static void main () { <decllist> <stmtlist> }

:= public static void main() {
<decl> ; <decllist> <stmtlist> }

And so on until you derive the program ...

:= public static void main() { String phrase = “baby fish should be fed ...” ; int numWords = 0; int pos = phrase.indexOf(“ “);
while (pos > 0) { numWords += 1;
... }

Determining if a Java program is syntactically correct

• Finite state machine (or determinisitic finite automaton - DFA) – to identify the words or tokens of the program
• Context-free grammar – to write the rules of the programming language
• LR Parsing determining if the program fits the rules – trying to derive the program.

• This area is known as Formal languages and Automata theory
Overview of 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!

Thanks to Students - Worked on JFLAP and Automata Theory Tools

• NPDA - 1990, C++, Dan Caugherty
• FLAP - 1991, C++, Mark LoSacco, Greg Badros
• JFLAP - 1996-1999, Java version Eric Gramond, Ted Hung, Magda and Octavian Procopiuc
• Pâté, JeLLRap, Lsys Anna Bilska, Jason Salemme, Lenore Ramm, Alex Karweit, Robyn Geer
• JFLAP 4.0 – 2003, Thomas Finley, Ryan Cavalcante
• JFLAP 6.0 – 2005-2008 Stephen Reading, Bart Bressler, Jinghui Lim, Chris Morgan, Jason Lee
• JFLAP 7.0 - 2009 Henry Qin, Jonathan Su
• JFLAP 8.0Beta – 2011-14 Julian Genkins, Ian McMahon, Peggy Li, Lawrence Lin, John Godbey
• JFLAP in OpenDSA – 2015 Sung-Hoon Kim and Martin Tamayo

Why Develop Tools for Automata?

| Textual | $(\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$
|         | $\delta = \{(q_0, b, q_3), (q_0, a, q_1), (q_1, a, q_1), (q_1, b, q_0), (q_0, b, q_1), (q_0, a, q_1)\}$
| Tabular |   | a | b |
|         |   | $q_0$ | $q_1$ | $q_3$ | $q_2$ |
| Visual  |   |   | a | a |
|         |   |   |   | b | a |
| Interactive |   | b | a |

DFA Example

• Build a deterministic finite automaton (DFA) to recognize even binary numbers with an even number of 1s.
• Only use symbols 0 and 1
• Binary numbers: 0, 1, 10, 11, 100, 101, 110, 111, ...
• When is a binary number an even number?
  • Ends in 0
• Which strings should be accepted?
  • 11010, 10010, 1111, 10100

| No, odd | Yes | No, ends | Yes |
| no. of 1’s | In 1 |
Build with JFLAP

Simulation on 1101010

Simulation on 1101010

Simulation on 1101010

Simulation on 1101010
Accepts Input! 1101010

Add meaning to states!

odd number of 1's
only one 0

odd number of 1's
only one 0
even number of 1's, ends in 1
Example: Build an NFA for valid integers

- Example:
  - Valid integers \{-3, 8, 0, 456, 13, 500, \ldots\}
  - Not valid: \{006, 3-6, 4.5, \ldots\}

Example: NFA for all valid integers
NFA annotated and shortcut

- Shortcut: [1-9] on labels

```
\[ A \rightarrow a A b c \\
A \rightarrow a b b c c \\
B X \rightarrow \lambda \\
B b \rightarrow b B \\
B c \rightarrow D \\
D X \rightarrow E X c \\
D b \rightarrow b D \\
D c \rightarrow c D \\
a E \rightarrow a B \\
b E \rightarrow E b \\
c E \rightarrow E c 
```

Back to Recognizing whether a Java Program is syntactically correct or not...

- You would need a DFA to recognize all valid words in a program
  - An integer
  - A variable name
  - All keywords
  - All special symbols ; + - ( ) { }
  - etc

Another Example: Grammar

- Grammar – set of replacement rules to define a language
- Previously looked at grammar for Java (very small part of it!)
- Grammar for a formal language (simpler)
- Consider representing underlined words in a text file (to be interpreted later):
  - cookie&&&&&______cookie
    & = go back one

Grammar for $a^n b^n c^n$

- Unrestricted grammar
- Generates strings with an equal number of a’s, b’s, c’s
- a’s first, then b’s, then c’s
- Example strings can derive:
  abc
  aabbbc
  aaabbbccc
  aaaaabbbbbcccccc
  ...

...
Example Derivation for aabbcc

S → AX  
→ aAbcX  
→ aAbBcbcX

rule: S -> AX  
rule: A -> aAbc  
rule: A -> aBbc

NOTE: We have generated the correct symbols, aabbcc, but they are in the wrong order!
Example Derivation for aabbcc

\[
\begin{align*}
S & \rightarrow AX \\
& \rightarrow aAbcX \\
& \rightarrow aaBbcbcX \\
& \rightarrow aabBcbcX \\
& \rightarrow aabDbcX \\
& \rightarrow aabbcDX
\end{align*}
\]

rule: \( S \rightarrow AX \)

rule: \( A \rightarrow aAbc \)

rule: \( A \rightarrow aBbc \)

rule: \( Bb \rightarrow bB \)

rule: \( Bc \rightarrow D \)

rule: \( Db \rightarrow bD \)

\[
\begin{align*}
& \rightarrow aAbcX \\
& \rightarrow aaBbcbcX \\
& \rightarrow aabBcbcX \\
& \rightarrow aabDbcX \\
& \rightarrow aabbcDX
\end{align*}
\]

rule: \( Bc \rightarrow D \)

rule: \( Db \rightarrow bD \)

rule: \( Dc \rightarrow cD \)

Note: the D absorbed the c!

Eventually \( \Rightarrow \) \( aabbcc \)
We could have done this derivation of aabbcc with JFLAP.

Now let’s see how JFLAP visualizes this derivation with a “parse tree”.
Note all letters there, but wrong order: aabcbc

Absorb the “c”
Spit out the “c” at the right end
Absorb second “c”

Spit the “c” out at right end
What else can JFLAP do?

- Create other machines
  - Moore and Mealy
  - Pushdown Automaton
  - Turing machine

- Parsing of grammars
  - regular, context-free grammars
  - Unrestricted grammar

- Conversions for proofs
  - NFA to DFA to minimal DFA
  - NFA $\leftrightarrow$ regular expression
  - NFA $\leftrightarrow$ regular grammar
  - CFG $\leftrightarrow$ NPDA

JFLAP - L-Systems

- L-Systems may be used to model biological systems and create fractals.
- Similar to Chomsky grammars, except all variables are replaced in each derivation step, not just one!
- Commonly, strings from successive derivations are interpreted as strings of render commands and are displayed graphically.
Add second T rule
L-Systems

The same stochastic L-system, rendered 3 different times all at the 9th derivation.
Conclusions From Study

- Results of Study showed
  - All the faculty used JFLAP in their courses, mostly for homework, some in lecture
  - Students had a high opinion of JFLAP
  - Majority of students felt access to JFLAP
    - Made learning course concepts easier
    - Made them feel more engaged
    - Made the course more enjoyable
  - Over half the students used JFLAP to study for exams
  - Over half the students thought time and effort using JFLAP helped them get a better grade.

Now a few tips if you ever write educational software...
Make your tool as interactive as possible – but not too tedious!

- User shouldn’t type everything
- Sometimes select
- Example: DFA to regular expression in JFLAP

Allow user to proceed on if they got it

- Complete the rest for them
- Complete parts for them

Avoid Too Many Pop up windows

- OLD JFLAP LR PARSE TOOL

Add Pause/Checkpoint questions

- Allow for pause to think about what comes next
- Undo/go back
- Pop up a quiz question to see if the user understands what he/she just did
  - JHAVE tool does this
  - Can integrate into ebooks
What can make the tool more useable?

- Annotations on states
- Multiple run window
  - Develop test data
  - Easier for grading
- General definitions
  - FA – recognize one or more symbols
  - NPDA – pop or push 0 or more symbols
- Batch processing

Naming your software

What is a “good” name for your tool?

- Algorithm Animation tool

Jawaa name is not unique

JAWAA name is not unique

How popular is JAWAA?

How popular is JAWAA?
FLAP

• Formal Languages and Automata Package

• 1996 – converted to Java

• FLAP -> JFLAP

Much more than Google Analytics Forums, Blogs, Course websites

Newest 'jflap' Questions - Stack Overflow
stackoverflow.com/questions/tagged/jflap

We can use small letters for terminals and caps for non-terminals in JFLAP while entering grammar. But this restricts to only 26 options. Can we have more ...

Blog: Recent posts - JFLAP
jflap.wikia.com/wiki/Blog


CS 301: Using JFLAP
www.cs.colostate.edu/~maesey/Teaching/.../JFLAP/gettingstarted.html

This course uses the JFLAP package. According to the JFLAP website, JFLAP is a package of graphical tools which can be used as an aid in learning the basic ...

PDF JFLAP Startup

Download JFLAP and the files referenced in this book from www.jflap.org to get started. JFLAP is written in Java to allow it to run on a range of platforms.
JFLAP is free

www.jflap.org

JFLAP tutorial

Outline

• Introduction
• CS Concepts Come Alive
  • Alice Programming Language
  • Algorithm Visualization
  • Automata Theory with JFLAP
  • Additional Ways to Engage with CS
• Diversity Efforts

Middle School students sorting themselves with Bubblesort
Interaction in Class – Props
Passing “Parameters” in Class

• Pass by reference – throw frisbee

• Pass by value – throw copy of frisbee

• Pass by const reference – throw “protected” frisbee

What happens when your hobby and your career collide?

It is now time for engaging students with edible CS

Automata Theory
Interaction in Class – Props
Edible Turing Machine

• TM for $f(x) = 2x$ where $x$ is unary

• TM is not correct, can you fix it? Then eat it!

• States are blueberry muffins
Students building DFA with cookies and icing

Outline

• Introduction
• CS Concepts Come Alive
  • Alice Programming Language
  • Algorithm Visualization
  • Automata Theory with JFLAP
  • Additional Ways to Engage with CS
• Diversity Efforts

Success - Alice Excites 4th-6th Grade Girls

• Duke Femmes Event, April 07
• 60 girls – 4 groups of 15
• Taught them Alice for an hour
• Handout to take home
• Event again in 2008, almost every year since

Adventures in Alice Programming

www.cs.duke.edu/csed/alice/aliceInSchools

• 2-week Teacher workshops
  • Over 200 teachers, middle school, high school, some elementary
  • First week Teach Alice, Practice
  • Second week - Develop Lesson Plans
  • All disciplines: math, science, history, language arts, foreign language, art, music, business
  • Summers 2008-2015, funding for lodging
• Main Sites:
  • Duke University, Durham, NC
  • Charleston/Columbia, SC
  • San Jose, CA (started 2014)
Curriculum Materials
www.cs.duke.edu/csed/alice/aliceInSchools

• Over 90 tutorials available for free
• Beginner, advanced, challenges, projects
• Paper handouts and video
• Over 200 Teacher lesson plans
  • Organized by discipline and grade level

Tutorial for Adventure Game –
Find objects in order

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

Science Example
How a volcano is formed

What a 6th grader can do with Alice
- teacher Chari Distler

No Superheros in Alice
How Visible are Notable Women in Computer Science?

- Pondered this question in early 2012
- Looked at Wikipedia
  - The internet encyclopedia
  - Who writes those pages?
  - Why did some notables have pages and others not?
- Turing Award Winners
  - Only two women at that time

Fran Allen

- School teacher – got a job at IBM
- Compilers and Optimization Technology
- IBM Fellow – First Women
- Turing Award (2006) – First Woman
- The Turing Award was announced on Feb. 21, 2007
  - Her Wikipedia page was created on...
    - Feb. 6, 2007
  - On Feb 21, 2007 the Turing Award was added to her Wikipedia page.

Here is that first page for Fran Allen

**Frances E. Allen**
From Wikipedia, the free encyclopedia

Frances Allen has made outstanding contributions to the field of programming languages for more than forty-five years, and her work has significantly influenced the wider computer science community. Ms. Allen is a pioneer in the field of optimizing compilers. Her achievements include seminal work in compilers, code optimization, and parallelization. In the early 1980s, she formed the Parallel TRAnslating (PTran) group to study the issues involved in compiling for parallel machines. The group was considered one of the top research groups in the world working with parallelization issues. Her work on these projects culminated in algorithms and technologies that form the basis for the theory of program optimization and are widely used in today's commercial compilers throughout the industry. Ms. Allen's influence on the IBM community was recognized by her appointment as an IBM Fellow, the first woman to receive this recognition. She was also president of the IBM Academy of Technology. The Academy plays an important role in the corporation by providing technical leadership, advancing the understanding of key technical areas and fostering communications among technical professionals.

In 1967, Ms. Allen was inducted into the WITI Hall of Fame. Ms. Allen retired from IBM in 2002.

Three days later...

**Frances E. Allen**
From Wikipedia, the free encyclopedia

This article has not been added to any categories. Please help out by adding categories to it so that it can be listed with similar articles.
Turing Award Announced and added to her page

In 1997, Ms. Allen was inducted into the WITI Hall of Fame. Ms. Allen retired from IBM in 2002. Early 2007, she became the first woman to win the A.M. Turing Award.

### A. M. Turing Award laureates

- Alan Perlis (1956)
- Maurice Vincent Wilkes (1957)
- Richard Hamming (1966)
- Marvin Minsky (1969)
- James H. Wilkinson (1970)
- John McCarthy (1971)
- Edsger W. Dijkstra (1972)
- Charles Bachman (1973)
- Donald Knuth (1974)
- Allen Newell / Herbert A. Simon (1975)
- Michael O. Rabin / Dana Scott (1976)
- John Backus (1977)
- Robert W. Floyd (1978)
- Kenneth E. Iverson (1979)
- Tony Hoare (1980)
- Edgar F. Codd (1981)
- Stephen Cook (1982)
- Ken Thompson / Dennis Ritchie (1983)
- Niklaus Wirth (1984)
- Richard Karp (1985)
- John Cocke (1987)
- Ivan Sutherland (1988)
- William Kahan (1989)
- Fernando J. Corbató (1990)
- Robin Milner (1991)
- Butler Lampson (1992)
- Douglas Engelbart (1996)
- Jim Gray / Fred Brooks (1997)
- Andrew Yao (2000)
- Ola-Johan Dahl / Kristen Nygaard (2002)
- Ron Rivest / Adi Shamir / Leonard Adleman (2002)
- Peter Norvig (2006)
- Frances E. Allen (2008)

Categories: Turing Award laureates

In the next three days

• Over 30 edits, added awards, boards

### Barbara Liskov

- Turing Award (2008)
- Currently Institute Professor at MIT
- One of the first women to get a PhD in CS in the U.S. (1968) – thesis on chess
- Research
  - Venus operating system
  - Design and Implementation of CLU
  - Argus, high-level language for distributed programs
  - Thor, object oriented database system
- Many awards – NAE, AAAS, Fellow ACM
- Had a Wikipedia page since before Turing Award
What about other Notable Women in Computer Science?

• ACM Fellows
  • Few women
    • 1994 first year over 130 Fellows
      • 9-12 were women? Less than 10%
  • About 20-50 Fellows per year
  • 2014 – 47 fellows, 6-8 women
• Noticed few of Women had Wikipedia pages
Investigate New CRA-W Project

- Write Wikipedia pages for Notable women in Computing
- How hard is it to write a Wikipedia page?
  - Lots of rules you have to follow
- Another area with few women
  - 2013 study – 16% of Wikipedia writers are female

Some Rules in Writing Wikipedia Biography pages

- You cannot write your own page!
- Neutral point of view
- Person must be notable
- No original research
  - Must write only facts and reference them
  - Must be verifiable
  - Do not plagiarize – write in your own words
- Regard for subject’s privacy
  - NOT A TABLOID!

Wrote a Wikipedia page

- We had no idea what we were doing....
- At a CRA-W Board meeting in April 2012

- Who to write?
  - Female Turing Award winners had pages
  - All two of them

Mary Jane Irwin

- Professor at Penn State University
- VLSI Architecture and Automated Design
- Board Level Designs
  - Arithmetic Cube, MGAP, and SPARTA
- Architecture, Logic & Circuit Design Tools
  - ARTIST, PERFLEX, LOGICIAN, DECOMPOSER
- Awards
  - National Academy of Engineering
  - ACM Fellow, IEEE Fellow
- No Wikipedia page!
In writing her Wikipedia page,

Here is what happens when you don’t know what you are doing...

Wrote a Guide on How to Write Wikipedia Biography
www.cs.duke.edu/csed/wikipedia

Our Database of Notable Women in CS
• Over 300 women
• Why notable
• Status of their Wikipedia page
• Forms for adding women and updating status
3 female Turing Award Winners
6 women Eniac Programmers
Anita Borg
Grace Hopper
ACM Fellows
Other

Thank You

• Questions?