First Bytes Collaborative Workshop for Computer Science Teachers

J Strother Moore
Chair
Department of Computer Sciences
University of Texas at Austin

July, 2008
Welcome to UT Austin CS

1. Stanford
2. Berkeley
3. MIT
4. CMU
5. Illinois
6. Cornell
6. Princeton
9. UT Austin
9. Georgia Tech
13. UCLA
13. Michigan
13. Maryland
16. Harvard
16. Columbia
16. Penn
18. Yale
18. Yale
19. Rice
20. Rice
20. Rice

11. Wisconsin
11. Caltech
13. UCSD
16. UCSD
## Recent Awards to the Faculty

<table>
<thead>
<tr>
<th>Year</th>
<th>Award</th>
<th>Recipient</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>ACM Turing Award</td>
<td>Allen Emerson</td>
</tr>
<tr>
<td>2008</td>
<td>NSF Career Award</td>
<td>Kristen Grauman</td>
</tr>
<tr>
<td>2008</td>
<td>NSF Career Award</td>
<td>Vitaly Shmatikov</td>
</tr>
<tr>
<td>2007</td>
<td>National Academy of Engineering</td>
<td>Simon Lam</td>
</tr>
<tr>
<td>2007</td>
<td>National Academy of Engineering</td>
<td>J Moore</td>
</tr>
<tr>
<td>2007</td>
<td>IJCAI Computers and Thought Award</td>
<td>Peter Stone</td>
</tr>
<tr>
<td>2007</td>
<td>NSF Career Award</td>
<td>Adam Klivans</td>
</tr>
<tr>
<td>2007</td>
<td>NSF Career Award</td>
<td>Emmett Witchel</td>
</tr>
<tr>
<td>2006</td>
<td>ACM SIGARCH Maurice Wilkes Award</td>
<td>Doug Burger</td>
</tr>
<tr>
<td>2006</td>
<td>SIAM Linear Algebra Prize</td>
<td>Inderjit Dhillon</td>
</tr>
<tr>
<td>2006</td>
<td>ACM Distinguished Scientist Award</td>
<td>Kathryn McKinley</td>
</tr>
<tr>
<td>2006</td>
<td>NSF Career Award</td>
<td>Bill Mark</td>
</tr>
<tr>
<td>2006</td>
<td>NSF Career Award</td>
<td>Lili Qiu</td>
</tr>
<tr>
<td>2006</td>
<td>NSF Career Award</td>
<td>Yin Zhang</td>
</tr>
<tr>
<td>Year</td>
<td>Award Name</td>
<td>Recipient(s)</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>2005</td>
<td>ACM Software System Award</td>
<td>Bob Boyer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Matt Kaufmann</td>
</tr>
<tr>
<td></td>
<td></td>
<td>J Moore</td>
</tr>
<tr>
<td>2005</td>
<td>NSF Career Award</td>
<td>William Cook</td>
</tr>
<tr>
<td>2004</td>
<td>ACM Software System Award</td>
<td>Simon Lam</td>
</tr>
<tr>
<td>2004</td>
<td>SIGCOMM Award</td>
<td>Simon Lam</td>
</tr>
<tr>
<td>2004</td>
<td>IEEE W. Wallace McDowell Award</td>
<td>Simon Lam</td>
</tr>
<tr>
<td>2004</td>
<td>Guggenheim Fellowship</td>
<td>David Zuckerman</td>
</tr>
<tr>
<td>2004</td>
<td>Sloan Fellowship</td>
<td>Peter Stone</td>
</tr>
<tr>
<td>2004</td>
<td>ONR Young Investigator Award</td>
<td>Peter Stone</td>
</tr>
<tr>
<td>2003</td>
<td>ACM Grace Murray Hopper Award</td>
<td>Steve Keckler</td>
</tr>
<tr>
<td>2003</td>
<td>NSF Career Award</td>
<td>Peter Stone</td>
</tr>
<tr>
<td>2002</td>
<td>Sloan Fellowship</td>
<td>Doug Burger</td>
</tr>
<tr>
<td>2002</td>
<td>Sloan Fellowship</td>
<td>Steve Keckler</td>
</tr>
<tr>
<td>2001</td>
<td>ACM Karlstrom Award</td>
<td>Nell Dale</td>
</tr>
<tr>
<td>2001</td>
<td>Sloan Fellowship</td>
<td>Lorenzo Alvisi</td>
</tr>
<tr>
<td>2001</td>
<td>NSF Career Award</td>
<td>Inderjit Dhillon</td>
</tr>
<tr>
<td>2000</td>
<td>Sloan Fellowship</td>
<td>Mike Dahlin</td>
</tr>
<tr>
<td>2000</td>
<td>NSF Career Award</td>
<td>Doug Burger</td>
</tr>
<tr>
<td>2000</td>
<td>NSF Career Award</td>
<td>Steve Keckler</td>
</tr>
<tr>
<td>Year</td>
<td>Award</td>
<td>Recipients</td>
</tr>
<tr>
<td>------</td>
<td>------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>1999</td>
<td>CADE Herbrand Award</td>
<td>Bob Boyer, J Moore</td>
</tr>
<tr>
<td>1998</td>
<td>ACM Kanellakis Award</td>
<td>Allen Emerson</td>
</tr>
<tr>
<td>1998</td>
<td>NSF Career Award</td>
<td>Lorenzo Alvisi</td>
</tr>
<tr>
<td>1998</td>
<td>NSF Career Award</td>
<td>Mike Dahlin</td>
</tr>
</tbody>
</table>
Job Data


Look at CNS Career Fair Data
But CS Is Not Just About Jobs

Computer science is the transformative science of our age.

It is also one of the deepest intellectual challenges in science.
Some Fundamental Ideas

Computability: There are fundamental limits on what can be computed. Technology can change \textit{how fast} we can compute. Technology cannot change \textit{what can be computed}.

Incompleteness: Some problems cannot be solved by computation.
Universality: Many different programming languages are equivalent: Anything that can be computed can be described in each of them.

Programs-as-data: Programs are just data. Programs can write programs.

Algorithm: There are different ways to solve problems.
Complexity: Certain problems are strongly equivalent to others in the sense that if any problem in the class can be solved with a certain amount of work all the other problems can be solved with the same amount of work.
Some Questions We Ask

How do people learn? Can we write programs that learn?


How do people know things?
How do they use that knowledge to solve problems?

Is the complexity class “P” the same as the complexity class “NP”?

What problems can you solve faster if you had a source of perfect randomness?
If you knew the DNA of all the animals in the world, could you reconstruct the evolutionary tree of life?

How can we build a faster computer?

Is biology just a computation?

Can we model how proteins fold?

Can we design drugs for a particular individual?
The Perilous Pipeline

Figure 6. BS Production

Number of Degrees

Year

Figure 7. Newly Declared CS/CE Undergraduate Majors
Some of the Things We’re Doing

helped with the inclusion of AP CS in 4x4
First Bytes Workshop for CS Teachers
First Bytes Summer Camp
CS Roadshow
EL Alliance
CRA (Computing Research Association)
ACM Educational Policy Committee
National Center for Women in IT (NCWIT)
National Academy of Engineering
Something You Can Do

Break the stereotypes associated with Computer Science.

Computer science is not about *computers*!

Computer science is the study of *computation*. 
An Analogy

“I want to major in computer science because I like computers.”

is like saying

“I want to be a novelist because I like using word processors.”
Chuck

Chuck likes word processing. So he wants to be a novelist.

You see Chuck in school, turning out long manuscripts (with perfect margins, nice fonts, and no spelling mistakes).

Chuck enrolls in the English Department at UT Austin.
Is he on his way to being a successful novelist?
Answer: Who Knows?
Does Chuck have the skills it takes to be a good writer?

- Creativity
- Mastery of the language
- Insight into people
- Interesting life experiences
Computer science isn’t about computers. Liking computers is not particularly important to being a good computer scientist.

Computer science is the study of computation.
The Skills of a Computer Scientist

• creativity

• mathematics: precision, abstraction, composition, symbolism

• problem solving: decomposition, refinement

• communication and teamwork
• ability to learn constantly and forever
Chuck

Chuck likes word processing. So he wants to be a novelist.

You see Chuck in school, turning out long manuscripts (with perfect margins, nice fonts, and no spelling mistakes).

Chuck enrolls in the English Department at UT Austin.
Chuck

Chuck likes computer games. So he wants to be a CS major.

You see Chuck in school, constantly talking about and showing off his skills with computer games and system administration.

Chuck enrolls in CS UT Austin.
Is he on his way to being a successful computer scientist?
The Skills of a Computer Scientist

• creativity

• mathematics: precision, abstraction, composition, symbolism

• problem solving: decomposition, refinement

• communication and teamwork
• ability to learn constantly and forever
Goal: A Partnership

Our goal is to re-excite young people in computing.

We need your help.

We’re willing to work with you but you’ll need to tell us what to do.

We may know a lot about computing but
we don’t know very much about K-12 issues.

Together maybe we can make a dent in this issue.
Thank You for Your Precious Time

I know you have many things to do during the summer.

Thank you for taking time out to spend with us.