Welcome to

The Hardware/Software Interface

Instructors:

Gaetano Borriello and Luis Ceze

Gaetano Borriello

costruzioni.net

espresso.repubblica.it

At UW since 1988

- *PhD at UC Berkeley*
- *MS at Stanford*
- *BS at NYU Poly*

Research trajectory:

- *Integrated circuits*
- *Computer-aided design*
- *Reconfigurable hardware*
- *Embedded systems*
- *Networked sensors*
- *Ubiquitous computing*
- *Mobile systems*
- *Applications in developing world*

Luis Ceze

At UW since 2007

- *PhD at U Illinois*
- *MEng at U São Paulo*
- *BEng at U São Paulo*

Research areas:

- *Architecture*
- *Multiprocessors*
- *Parallelism*
- *Compilers*

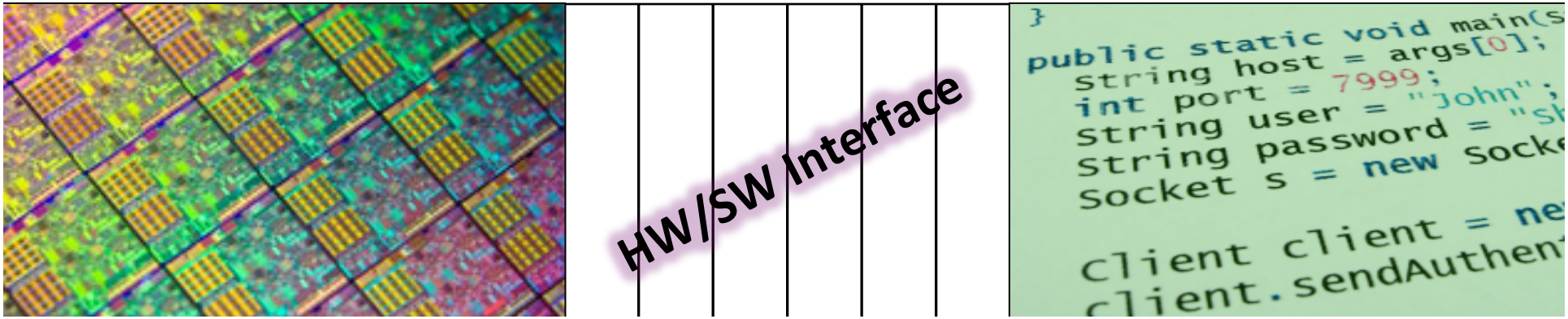
UW Computer Science & Engineering

Who are you?

- There are literally thousands of you
- We do not get to meet you face to face ☹
- But please fill our the course on-line survey so that we get to know a few things about you
- We'll report aggregate statistics at the end of the course

What is this class about?

- What is hardware? software?
- What is a hardware/software interface?
- Why do we need to understand this interface?



Recommended prerequisites

- **What we expect you to know**
  - Basics of binary numbers
    - $1001_2 + 0001_2 = 1010_2$
  - Binary logic operators: AND, OR, and NOT
    - A AND B is true if and only if A is true and B is true and false otherwise
  - Introductory programming in Java (or C)
    - if statements, loops, procedures/methods
- **What we expect you to have**
  - Access to a modern personal computer (Windows, MacOS, or Linux) on which you can install some software

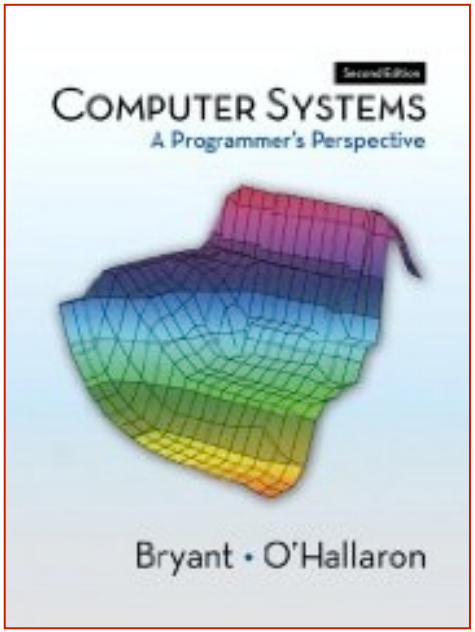
Course outcomes

- **Foundation: basics of computer programming (Java)**
- **Understanding of some of the abstractions that exist between programs and the hardware they run on, why they exist, and how they build upon each other**
- **Knowledge of some of the implementation details of these underlying abstractions**
- **Become more effective programmers**
  - More efficient at finding and eliminating bugs
  - Understand some of the many factors that influence program performance
  - Facility with a couple more of the many languages that we use to describe programs and data
- **Prepare for later classes in computing**

What we will have you do

- **Five programming assignments**
  - 1 to 2 weeks each
- **Some recommended pencil/paper problems**
- **We'll also point you to relevant sections of a textbook**
  - We are hoping to make the lectures self-contained
  - View the textbook as supplementary material for a second point-of-view

Reference texts



- **Computer Systems: A Programmer's Perspective, 2nd Edition**
  - Randal E. Bryant and David R. O'Hallaron
  - Prentice-Hall, 2010
  - http://csapp.cs.cmu.edu
  - Purchase direct from Pearson:
    http://www.mypearsonstore.com/bookstore/computer-systems-a-programmers-perspective-9780136108047
  - Purchase eBook from CourseSmart:
    http://www.coursesmart.com/0132130661/?a=1773944
  - Purchase print or Kindle edition from Amazon.com:
    http://www.amazon.com/Computer-Systems-Programmers-Perspective-2nd/dp/0136108040

- **A good C reference – any will do – lots of info on the web**
  - The C Programming Language (Kernighan and Ritchie)
  - C: A Reference Manual (Harbison and Steele)

Acknowledgments

- **The many TAs behind the scenes**
- **The previous Coursera instructors at UW for sharing their experiences with us**
- **The many instructors for the subject of this course who have shared their lecture notes – they deserve a lot of the credit, the errors are all ours**
  - CMU: Randy Bryant, David O'Hallaron, Gregory Kesden, Markus Püschel
  - Harvard: Matt Welsh (now at Google-Seattle)
  - UW: Peter Hornyack, Hal Perkins, John Zahorjan