The 2014 CS Teaching Academy An Introduction and Rationale

David Bernstein and Chris Mayfield

Department of Computer Science James Madison University

 $23 \ \mathrm{June} \ 2014$ 

## About This Presentation

- Typical undergraduate programs in computer science
- Summary of the new "AP CS Principles" course
- The rationale for a survey course
- CS101 at JMU
- The rationale for this workshop



# Characteristics of Graduates

- Technical understanding of computer science
- Familiarity with common themes and principles
- Appreciation of the interplay between theory and practice
- System-level perspective
- Problem solving skills
- Project experience
- Commitment to life-long learning
- Commitment to professional responsibility
- Communication and organizational skills
- Awareness of the broad applicability of computing
- Appreciation of domain-specific knowledge

### Knowledge Areas and Core Lecture Hours

	2013		2008	2001
Knowledge Area	Tier 1	Tier 2	Core	Core
Algorithms and Complexity	19	9	31	31
Architecture and Organization	0	16	36	36
Computational Science	1	0	0	0
Discrete Structures	37	4	43	43
Graphics and Visual Computing	2	1	3	3
Human Computer Interaction	4	4	8	8
Security and Information Assurance	2	6	0	0
Information Management	1	9	11	10
Intelligent Systems	0	10	10	10
Networking and Communications	3	7	15	15
Operating Systems	4	11	18	18
Parallel and Distributed Computing	5	10	0	0
Programming Languages	8	20	21	21
Software Development Fundamentals	43	0	47	38
Software Engineering	6	21	31	31
Systems Fundamentals	18	9	0	0
Social Issues and Professional Practice	11	5	16	16
	164	142	290	280

# Current Core Courses ( $\approx 500$ Lecture and Lab Hours)

- Programming Fundamentals (CS139/CS149)
- Advanced Programming (CS159)
- Data Structures and Algorithms (CS240)
- Discrete Mathematics I (CS227)
- Discrete Mathematics II (CS228)
- Technical Communication (CS260)
- Software Engineering (CS345)
- Computer Organization (CS350)
- Programming Languages (CS440)
- Operating Systems (CS450)
- TCP/IP Networks (CS460)
- Database Design and Application (CS474)

# Changes in the Future

#### • Possible Changes:

Reorganization of the systems core (CS350, CS450, CS460) Required course on social issues and professional practice Required survey course (CS101, more on this shortly) Required project course(s)

• Very Unlikely to Change:

Reduction in breadth/"Gen Ed" (JMU is a liberal arts university)

# CSP Curriculum Framework

#### http://www.csprinciples.org/home/about-the-project

 $\begin{array}{ll} \mbox{Big Ideas} \rightarrow & \\ \mbox{Enduring Understandings} \rightarrow & \\ \mbox{Learning Objectives} \rightarrow & \\ \mbox{Essential Knowledge Statements} \end{array}$ 

Computational Thinking Practices  $\rightarrow$ Performance Tasks (including rubrics)

Hansen, Edmund. 2011. Idea-based learning: A course design process to promote conceptual understanding. Sterling, VA: Stylus.

# 7 Big Ideas

- Computing is a <u>creative</u> activity.
- 2 <u>Abstraction</u> reduces information and detail to facilitate focus on relevant concepts.
- **<u>③</u>** Data and information facilitate the creation of knowledge.
- Algorithms are used to develop and express solutions to computational problems.
- Programming enables problem solving, human expression, and creation of knowledge.
- <u>The Internet</u> pervades modern computing.
- Computing has **global impact**.

# CT Practices (and assessment)

- Connecting computing
- ② Creating computational artifacts
- Abstracting
- Analyzing problems and artifacts
- Ommunicating
- 6 Collaborating
- **1** Investigate Bits to Information to Knowledge
- Explore Implications of Computing Innovations
- Create Applications from Ideas (programming)

## Our Perspective on the Current "CS Principles"

• It Does...

Expose students to some aspects of computational thinking Promote the use of computing/computers

• It May Not... (Survey Results Coming Soon) Help students understand/appreciate computer science Be aligned with current B.A./B.S. programs Prepare students to study computer science in college

> Just ask Google: CS AP Principles (is that a bug or an Easter egg?!)

### Disclaimer



- Overall, the CSP community is doing a really great job!
- A lot of talented people are working hard to design CSP
- We have a unique<sup>1</sup> approach we'd like to share with you

<sup>&</sup>lt;sup>1</sup>Unique to CS Principles, but not to science education

The Rationale for a Survey Course

What is a Survey Course?

#### Definition

A survey course is a course that provides an overview of the principal topics of a broad field of knowledge.

# Examples of Survey Courses

#### • Introduction to Economics

Provides an overview of microeconomics (the theory of the firm, the theory of the consumer, and the theory of individual markets), macroeconomics (the study of the performance and structure of economies), and econometrics (the application of mathematics and statistics to economic data).

### • Survey of English Literature

A study of significant writings in English literature (usually drawn from a particular period).

#### • Introduction to Communication

Examines the elements affecting communication (written, spoken, and non-verbal) at the individual, small group, and public communication levels.

#### • Introductory Physics

A brief introduction to mechanics, energy, momentum conservation, thermodynamics, fluids, electricity, magnetism, and optics.

### The Rationale

#### • Majors/Minors:

Provides context for other courses Provides a foundation for other courses Allows us to cover "smaller" topics (i.e., ones that don't require an entire course)

#### • Prospective Majors/Minors:

Provides a low risk way to "test the waters"

#### • Non-Majors:

Provides a way to understand what computer scientists think about and how they think

#### CS101 at JMU

## CS101 Topics

- Data Storage (CS227)
- Architecture (CS350)
- Operating Systems (CS450)
- Networking (CS460)
- Security (Elective)
- Algorithms (CS139/149)
- Programming Languages (CS430)
- Software Engineering (CS345)
- Data Structures (CS240)
- Databases (CS474)
- Artificial Intelligence (Elective)
- Theory of Computing (Elective)
- Impact on Society (Elective)

## Some Observations

• The topics in CS101 are not covered in the same order as in our curriculum

It is easier to cover topics in the "correct" order in CS101

- Some topics in our core curriculum are not covered in CS101 There isn't enough time CS101 is designed for both majors and non-majors
- Some topics in CS101 are not in our core curriculum Non-majors find these topics interesting All majors don't need a full course but do need some exposure We do offer elective courses on these topics

### CS101 Design

# Example Learning Objective

Use abstraction and decomposition when reasoning about complex systems and problems.

### Lab01 example



"F1" is an abstract tool

### Lab02 example



"Half adder" is an abstract tool

### Lab03 example



"CPU instructions" are abstract tools

### Essence of CS101

The point is NOT:

- Become an expert at Light-Bot programming
- Be able to understand/design complex circuits
- Program a computer in machine language

The point is:

- Learn how to think like a computer scientist
- Sample what you will learn in future courses
- Develop new computing skills (e.g., Linux)

# Focus of CS Academy 2014

#### • Our goals...

Provide you with a broad understanding of the discipline Prepare you to teach computer science as a distinct field Focus on core concepts/practices, not just programming

#### • What you will receive...

New ideas, new materials, and new friends Support in your professional development Great food and lots of free stuff! (daily)

In short, we want you to experience CS101...in one week! (That's why it's called CS501 for you.)