ACM Undergraduate Curricular Guidance in Computer Science: The First Two Years

Elizabeth K. Hawthorne, Cara Tang, Cindy S. Tucker, Teresa Moore, Lambros Piskopos, Christian Servin

Overview of the Guidance

- **Updates the ACM *Computing Curricula 2009: Guidelines for Associate-Degree Transfer Curriculum in Computer Science***
- **Includes contemporary cybersecurity concepts**
- **Uses the ACM *Computer Science Curricula 2013 (CS2013)* as a starting point**

Driving Factors

- The dated *ACM Computing Curricula 2009: Guidelines for Associate-Degree Transfer Curriculum in Computer Science*
- The release of CS2013
- An expressed need by community college educators at SIGCSE 2015
- A projected multi-year global cybersecurity workforce shortfall
- A top U.S. priority to build a highly capable cybersecurity workforce

Process

Under the auspices of the ACM Education Board, the Committee for Computing Education in Community Colleges (CCECC):

Phase 1: Develop a initial draft called StrawDog June 2016

- Convened a task force of community college educator to develop the initial draft of the updated guidance
- Administered a survey to a global audience of CS educators on CS2013 foundational knowledge deemed appropriate for the first two years of a computer science education
- Administered a survey to a global audience of CS educators on contemporary cybersecurity concepts deemed appropriate for the first two years of a computer science education
- Built the guidance on a framework of learning outcomes
- Included meaningful evaluation metrics
- Released for public review and comment

Phase 2: Develop a second draft called IronDog September 2016

- Incorporate feedback on StrawDog
- Release for public review and comment

Phase 3: Final Version Q1 2017

- Incorporate feedback on IronDog
- Release for public review and comment

Student Learning Outcomes (SLOs)

- ✓ **Computer Science competencies**
- ✓ **Cybersecurity competencies**
- ✓ **Assessment rubric for each outcome**

Sample SLOs

Use a programming language to implement, test, and debug algorithms for solving simple problems.

Differentiate between prevailing types of operating systems, such as networked, mobile, real-time, and distributed.

Describe user responsibilities related to the handling of information in both personal and enterprise computing.

Analyze an existing software implementation and make suggestions to improve security in its design.

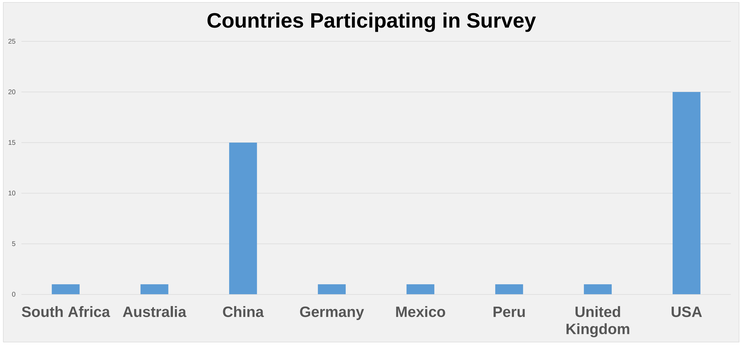
Discuss potential threats to operating systems and the security features designed to guard against them.

Explain global and local laws/regulations related to cyber law and computing ethics.

SLOs with Assessment Rubric

Learning Outcome	Below Expectations	Meets Expectations	Exceeds Expectations
Utilize defensive programming methods, such as input validation and type checking when writing programs.	Define various defensive programming methods such as input validation and type checking	Utilize defensive programming methods, such as input validation and type checking when writing programs.	Write programs that use defensive programming methods, like input validation and type checking to solve a business problem.
Implement a simple client-server socket-based application.	Identify the components of a simple client-server socket-based application.	Implement a simple client-server socket-based application.	Integrate a simple client-server socket-based application within a server program to exchange data with a client.
Explain recent and historical legislation related to digital privacy, unlawful access, digital piracy, cyber defense, and computing ethics.	Identify recent and historical legislation related to digital privacy, unlawful access, digital piracy, cyber defense, and computing ethics.	Explain recent and historical legislation related to digital privacy, unlawful access, digital piracy, cyber defense, and computing ethics.	Assess the consequences of violating any of the legislation related to digital privacy, unlawful access, digital piracy, cyber defense, or computing ethics.

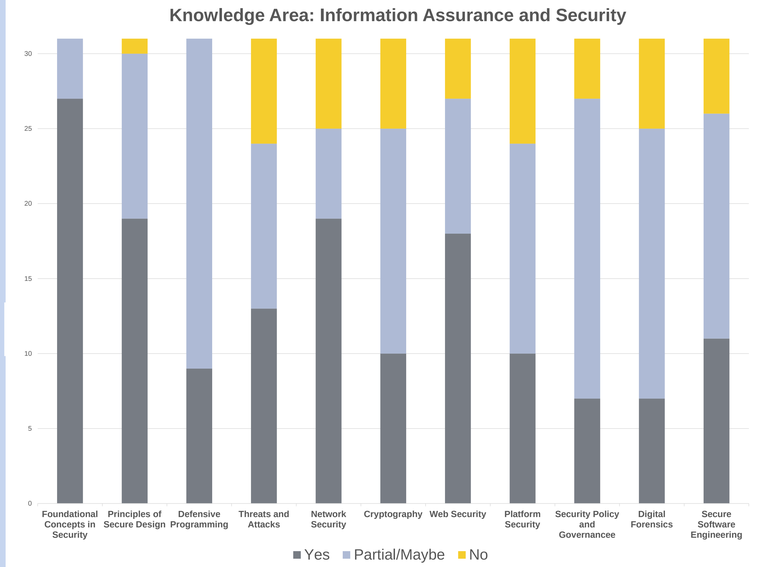
Global Influence



CS2013 KAs in StrawDog

- 15 of 18 CS2013 KAs are appropriate, in some part and at some level, in the first two years of computer science
- Three CS2103 KAs are not included: Intelligent Systems, Platform-based Development, and Parallel and Distributed Systems

Sample Survey Question



Contact Us

For additional information or to help review CS-Cyber StrawDog or IronDog, visit our website at

http://ccecc.acm.org/