ACM Computer Science Curricular Guidance for Associate-Degree Transfer Programs with Infused Cybersecurity

Panel: Elizabeth Hawthorne, Cara Tang, Cindy Tucker, Christian Servin, Markus Geissler

http://ccecc.acm.org
Agenda

Introduction & Milieu
Overview of Guidance
Cybersecurity Infused Learning Outcomes
CAE2Y & AP CS-A Mappings
Program Examples and Call for More
Staying in Contact
Question and Answers
Committee for Computing Education in Community Colleges

Celebrating 40++ years of service to computing education
Standing committee of the ACM Education board for 25++ years

Global Mission: Serve and support community and technical college educators in all aspects of computing education
Engage in curriculum and assessment development, community building, and advocacy in service to this sector of higher education
CS-Transfer Guidelines Background

ACM Curricular Guidance – [www.acm.org/education/](http://www.acm.org/education/)

2009: Guidelines for Associate-Degree Transfer Curriculum in Computer Science

2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science – CS 2013

  - New knowledge area: Information Assurance and Security (IAS)

2015: BoF @ SIGCSE: Perspectives on How CS 2013 Influences Two-Year College Programs – Standing room only!

2015: Joint task force on Cybersecurity Education formed – ACM, IEEE-CS, AIS-SEC, CEP

2015: NSF C5 Project – Catalyzing Computing and Cybersecurity in Community Colleges

2016: Computer Science for All – U.S. Government initiative
  ([www.whitehouse.gov/blog/2016/01/30/computer-science-all](http://www.whitehouse.gov/blog/2016/01/30/computer-science-all))
CS-Transfer Guidelines Background

**Nov 2015**: CS-Cyber Task Group formed

Divide CS 2013 knowledges areas (KAs) into 3 clusters, form 3 teams
   Team leads: Teresa Moore, Lambros Piskopos, Christian Servin

For each CS2013 knowledge unit (KU): Appropriate for associate-degree level?

Draft learning outcomes for each KU
   Sources: CS 2013, NSA CAE2Y, NICE Framework, IT 2017 v0.51, Bloom’s Taxonomy

**Mar 2016**: SIGCSE workshop developing learning outcomes & assessment rubric

**June 2016**: StrawDog released; 2 surveys for input
   Over 50 feedback responses from 8 different countries
CS-Transfer Guidelines Background

**July 2016**: Poster at ITiCSE, Arequipa, Peru

**Oct 2016**: IronDog released

**Jan – Feb 2017**: Final input incorporated
   - KAs, KUs, learning outcomes, assessment rubric, and Bloom’s levels reviewed, tweaked, and finalized

**Mar 2017**: Pre-release of Final version @ SIGCSE 2017, Seattle
   ACM Education Board Endorsed

**June 2017**: Final version release @ 3CS 2017
   Available ACM Digital Library
Acknowledgements

Team Leaders and Task Force Members
- Prof. Lambros Piskopos, Wilbur Wright College, IL, Team Leader
- Dr. Markus Geissler, Cosumnes River College, CA, Team Leader
- Prof. Kimberly Bertschy, Northwest Arkansas Community College, AR
- Prof. Colleen Case, Schoolcraft College, MI
- Prof. Rafael Escalante, El Paso Community College, TX
- Dr. Becky Grasser, Lakeland Community College, OH
- Prof. Charles Hardnett, Gwinnett Technical College, GA
- Prof. Amardeep Kahlon, Austin Community College District, TX
- Prof. James Kolasa, Bluegrass Community and Technical College, KY
- Dr. Shamsi Moussavi, MassBay Community College, MA
- Prof. Pam Schmelz, Ivy Tech Community College, IN
- Prof. Melissa Stange, Lord Fairfax Community College, VA
- Prof. Khallai Taylor, Miami-Dade College, FL
- Prof. Carole Tharnish, Northeast Community College, NB

Other Contributors
- Dr. Anne Applin, Southern Maine Community College, ME
- Prof. Bryce Barrie, Saskatchewan Polytechnic, Canada
- Prof. Michael Bauer, Leeward Community College, HI
- Prof. Paul Dadosky, Ivy Tech Community College, IN
- Prof. Andrea DeMott, Ohio University, OH
- Dean Jamie Edwards, Wytheville Community College, VA
- Dr. Larry Forman, San Diego City College, CA
- Prof. Guy Garrett, Gulf Coast State College, FL
- Prof. Dianne Hill, Jackson College, MI
- Dr. Nancy Jones, Coastline Community College, CA
- Prof. Marc Nester, Wytheville Community College, VA
- Dr. Dean Nevins, Santa Barbara City College, CA
- Dr. Michael Posner, Villanova University, PA
- Prof. Kristopher Roberts, Ivy Tech Community College, IN
- Prof. Barry Sullens, Ivy Tech Community College, IN
- Prof. Robert Surton, Columbia Gorge Community College, OR
Differences from CS2013

- 17 of 18 KAs included
  - 1 KA not included: Intelligent Systems
- 1 KA name change:
  - IAS Information Assurance and Security → CYB Cybersecurity
- Various KUs included for each KA
- Learning Outcomes updated
  - Utilize Bloom’s Revised Taxonomy
- Assessment rubric added for every learning outcome
- No topics
Why Learning Outcomes Approach?

Focus is on **student achievement**

Supports modification of existing courses (easier to add outcomes than entire courses)

Also supports development of new courses

Avoids traditional body of knowledge focus on topics and contact hours that can grow unbounded as new technologies emerge

What topics are eliminated to make room for the new? (food fight)
3-Tiered Assessment Rubric

Every learning outcome has an assessment rubric

<table>
<thead>
<tr>
<th>Learning Outcome</th>
<th>Emerging Standard</th>
<th>Developed Standard</th>
<th>Highly Developed Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>CYB-15. Construct input validation and data sanitization in applications,</td>
<td>Implement simple input validation and data sanitization in applications. [Applying]</td>
<td>Construct input validation and data sanitization in applications, considering adversarial control of the input channel. [Creating]</td>
<td>Develop complex input validation and data sanitization in applications, considering adversarial control of the input channel. [Creating]</td>
</tr>
</tbody>
</table>
Body of Knowledge

17 knowledge areas
214 learning outcomes with assessment metrics
63 learning outcomes specific to cybersecurity
  25 in CYB knowledge area
  38 in other KAs
### Body of Knowledge

<table>
<thead>
<tr>
<th>Course</th>
<th>LOs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms and Complexity (AL)</td>
<td>17 LOs</td>
</tr>
<tr>
<td>Architecture and Organization (AR)</td>
<td>11 LOs</td>
</tr>
<tr>
<td>Computational Science (CN)</td>
<td>3 LOs</td>
</tr>
<tr>
<td>Cybersecurity (CYB)</td>
<td>25 LOs</td>
</tr>
<tr>
<td>Discrete Structures (DS)</td>
<td>34 LOs</td>
</tr>
<tr>
<td>Graphics and Visualization (GV)</td>
<td>5 LOs</td>
</tr>
<tr>
<td>Human-Computer Interaction (HCI)</td>
<td>6 LOs</td>
</tr>
<tr>
<td>Information Management (IM)</td>
<td>13 LOs</td>
</tr>
<tr>
<td>Networking and Communications (NC)</td>
<td>8 LOs</td>
</tr>
<tr>
<td>Operating Systems (OS)</td>
<td>13 LOs</td>
</tr>
<tr>
<td>Parallel and Distributed Computing (PD)</td>
<td>5 LOs</td>
</tr>
<tr>
<td>Platform-based Development (PBD)</td>
<td>No LOs</td>
</tr>
<tr>
<td>Programming Languages (PL)</td>
<td>10 LOs</td>
</tr>
<tr>
<td>Software Development Fundamentals (SDF)</td>
<td>19 LOs</td>
</tr>
<tr>
<td>Software Engineering (SE)</td>
<td>14 LOs</td>
</tr>
<tr>
<td>System Fundamentals (SF)</td>
<td>9 LOs</td>
</tr>
<tr>
<td>Social Issues and Professional Practice (SP)</td>
<td>22 LOs</td>
</tr>
</tbody>
</table>
Cybersecurity Infused Learning Outcomes

![Bar Chart]

- **AL**: 2
- **AR**: 1
- **CYB**: 25
- **GV**: 1
- **HCI**: 1
- **IM**: 4
- **NC**: 2
- **OS**: 3
- **PD**: 2
- **PL**: 3
- **SDF**: 4
- **SE**: 6
- **SF**: 2
- **SP**: 7

*Number of Cybersecurity-infused Learning Outcomes*
Mappings CS Transfer to Other Curriculum, Frameworks, and Classifications

NSA/DHS CAE2Y Knowledge Units
College Board AP Computer Science A
ACM Computer Science 2013 Guidance
Others
Percentage of CS Transfer Knowledge Area Learning Outcomes Mapped to CAE2Y Knowledge Units

- SP: SOCIAL ISSUES AND PROFESSIONAL PRACTICE: 50%
- SF: SYSTEM FUNDAMENTALS: 22%
- SE: SOFTWARE ENGINEERING: 21%
- SDF: SOFTWARE DEVELOPMENT FUNDAMENTALS: 58%
- PL: PROGRAMMING LANGUAGES: 10%
- PD: PARALLEL AND DISTRIBUTED COMPUTING: 0%
- OS: OPERATING SYSTEMS: 46%
- NC: NETWORKING AND COMMUNICATION: 63%
- IM: INFORMATION MANAGEMENT: 8%
- HCI: HUMAN-COMPUTER INTERACTION: 33%
- GV: GRAPHICS AND VISUALIZATION: 0%
- DS: DISCRETE STRUCTURES: 0%
- CYB: CYBERSECURITY: 84%
- CN: COMPUTATIONAL SCIENCE: 0%
- AR: ARCHITECTURE AND ORGANIZATION: 9%
- AL: ALGORITHMS AND COMPLEXITY: 12%
AP Computer Science A Topics Mapping

Percentage of CS Transfer Knowledge Area Learning Outcomes Mapped to Advanced Placement Computer Science A Topics

- SP: SOCIAL ISSUES AND PROFESSIONAL PRACTICE: 32%
- SF: SYSTEM FUNDAMENTALS: 0%
- SE: SOFTWARE ENGINEERING: 7%
- SDF: SOFTWARE DEVELOPMENT FUNDAMENTALS: 0%
- PL: PROGRAMMING LANGUAGES: 40%
- PD: PARALLEL AND DISTRIBUTED COMPUTING: 0%
- OS: OPERATING SYSTEMS: 0%
- NC: NETWORKING AND COMMUNICATION: 13%
- IM: INFORMATION MANAGEMENT: 0%
- HCI: HUMAN-COMPUTER INTERACTION: 0%
- GV: GRAPHICS AND VISUALIZATION: 0%
- DS: DISCRETE STRUCTURES: 0%
- CYB: CYBERSECURITY: 0%
- CN: COMPUTATIONAL SCIENCE: 0%
- AR: ARCHITECTURE AND ORGANIZATION: 36%
- AL: ALGORITHMS AND COMPLEXITY: 53%
Program Examples

Computer Science 2017 Program Example from El Paso Community College

Submitted: 2017-06-19

Description
This two-year Computer Science Field of Study prepares students to transfer directly into a Bachelor's Degree Program in Computer Science at a four-year institution. Students planning to enter the computer science field or the Associates Degree in Computer Programming or Telecommunications and Networking are advised to select electives which will apply to the academic course working that degree plan. The Computer Science Space Field of Study will provide a balanced program which will give the students a strong concentration in the computer programming area. It is highly recommended that students complete the Math and Physics sequence at the same institution. The student who plans to major in Computer Science may take some of the required courses as both Core Curriculum requirements and Field of Study requirements.

Program Website
https://www.epcc.edu/InstructionalPrograms/Pages/ComputerScience.aspx

Contact
## Program Examples (cont-d)

<table>
<thead>
<tr>
<th>Area</th>
<th>KU-Description</th>
<th>Learning Outcome</th>
<th>COSC 1436</th>
<th>COSC 1437</th>
<th>COSC 2336</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AL-02. Estimate time and space complexities for a given algorithm using Big-O notation. [Evaluating]</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AL-03. Contrast standard complexity classes. [Analyzing]</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AL-04. Analyze the performances of an algorithm with various input sizes. [Analyzing]</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AL-08. Implement common search algorithms, including linear and binary searches. [Applying]</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>AL: Algorithms and Complexity</td>
<td>Fundamental Data Structures and Algorithms</td>
<td>AL-10. Implement hash tables, including collision avoidance and resolution. [Applying]</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Area</td>
<td>KU-Description</td>
<td>Learning Outcome</td>
<td>COSC 1436</td>
<td>COSC 1437</td>
<td>COSC 2336</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------------------------</td>
<td>------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>SDF: Software Development Fundamentals Fundamental Programming Concepts</td>
<td>SDF-06. Create programs which use defensive programming techniques, including input validation, type checking, and protection against buffer overflow. [Creating]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SDF: Software Development Fundamentals Fundamental Programming Concepts</td>
<td>SDF-07. Create code in a programming language that includes primitive data types, references, variables, expressions, assignments, I/O, control structures, and functions. [Creating]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SDF: Software Development Fundamentals Fundamental Programming Concepts</td>
<td>SDF-08. Create a simple program that uses persistence to save data across multiple executions. [Creating]</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDF: Software Development Fundamentals Fundamental Programming Concepts</td>
<td>SDF-09. Create a simple program that uses recursion. [Creating]</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDF: Software Development Fundamentals Fundamental Data Structures</td>
<td>SDF-10. Create simple programs that include each of the following data structures: lists, stacks, queues, hash tables, graphs and trees. [Creating]</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDF: Software Development Fundamentals Development Methods</td>
<td>SDF-12. Investigate common coding errors that introduce security vulnerabilities, such as buffer overflows, integer errors, and memory leaks. [Applying]</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Program Examples (cont-d)

<table>
<thead>
<tr>
<th>Area</th>
<th>KU-Description</th>
<th>Learning Outcome</th>
<th>COSC 1436</th>
<th>COSC 1437</th>
<th>COSC 2336</th>
</tr>
</thead>
<tbody>
<tr>
<td>CYB: Cybersecurity</td>
<td>Foundational Concepts in Security</td>
<td>CYB-02. Illustrate through examples the concepts of risk, threats, vulnerabilities, attack vectors, and exploits, noting there is no such thing as perfect security.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>CYB: Cybersecurity</td>
<td>Foundational Concepts in Security</td>
<td>CYB-05. Explain the concepts of trust and trustworthiness related to cybersecurity. [Understanding]</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>CYB: Cybersecurity</td>
<td>Foundational Concepts in Security</td>
<td>CYB-07. Illustrate various ways to minimize privacy risks and maximize anonymity.</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CYB: Cybersecurity</td>
<td>Principles of Secure Design</td>
<td>CYB-12. Illustrate the security implications of relying on open design vs the secrecy of design. [Applying]</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>CYB: Cybersecurity</td>
<td>Defensive Programming</td>
<td>CYB-15. Construct input validation and data sanitization in applications, considering adversarial control of the input channel. [Creating]</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CYB: Cybersecurity</td>
<td>Defensive Programming</td>
<td>CYB-17. Implement programs that properly handle exceptions and error conditions. [Applying]</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>CYB: Cybersecurity</td>
<td>Defensive Programming</td>
<td>CYB-18. Examine the need to update software to fix security vulnerabilities. [Analyzing]</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CYB: Cybersecurity</td>
<td>Cryptography</td>
<td>CYB-21. Describe key terms in cryptology, including cryptography, cryptanalysis, cipher, cryptographic algorithm, and public key infrastructure. [Understanding]</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CYB: Cybersecurity</td>
<td>Cryptography</td>
<td>CYB-22. Use a variety of ciphers to encrypt plaintext into ciphertext. [Applying]</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Stay Involved with ACM CCECC

Follow us on Twitter @ACMccecc or Facebook ACMccecc

Google group: cc-ed-computing
   Open to anyone interested in computing education at community & technical colleges
   Post any questions / discussion items to have a conversation with the community

Bookmark our Website
CCECC.ACM.org