Curricular Guidance for Associate-Degree Transfer Programs in Computer Science with Contemporary Cybersecurity Concepts

ACM Committee for Computing Education in Community Colleges (CCECC)

Cara Tang, Elizabeth K. Hawthorne, Cindy S. Tucker, Christian Servin
cara.tang@pcc.edu, ehawthorne@ccecc.acm.org, cindy.tucker@kctcs.edu, cservin1@epcc.edu

Overview

- **Reflects influence derived from 17 of the 18 Knowledge Areas in the ACM/IEEE Computer Science Curricula 2013.**
- **Provides a significant update to the 2009 associate-degree computer science guidance by including contemporary cybersecurity learning outcomes.**
- **Contains a set of learning outcomes which primarily span the first three levels of the cognitive domain from Bloom's Revised Taxonomy.**
- **Includes a three-tiered assessment rubric with meaningful evaluation metrics for each student learning outcome.**

KAs of the Associate-Degree CS Transfer Curriculum

Algorithms and Complexity (AL)
Basic Analysis
Algorithmic Strategies
Fundamental Data structures and Algorithms
Basic Automata, Computability, and Complexity

Architecture and Organization (AR)
Digital Logic and Digital Systems
Machine Level Representation of Data
Assembly Level Machine Organization
Memory System Organization and Architecture

Computational Science (CN)
Introduction to Modeling and Simulation

Cybersecurity (CYB)
Foundational Concepts in Security
Principles of Secure Design
Defensive Programming
Threats and Attacks
Cryptography
Web Security
Secure Software Engineering

Discrete Structures (DS)
Sets, Relations, and Functions
Basic Logic
Proof Techniques
Basics of Counting
Graphs and Trees
Discrete Probability

Graphics and Visualization (GV)
Fundamental Concepts

Human Computer Interaction (HCI)
Foundations
Designing Interaction

Information Management (IM)
Information Management Concepts
Database Systems
Data Modeling

Networking and Communication (NC)
Introduction
Networked Applications

Operating Systems (OS)
Overview of Operating Systems
Operating System Principles
Concurrency
Memory Management
Security and Protection
Virtual Machines
Device Management

Parallel and Distributed Computing (PD)
Parallelism Fundamentals
Communication and Coordination

Platform-based Development (PBD)
Web Platforms
Mobile Platforms
Industrial and Robotic Platforms
Gaming Platforms
Cloud and IoT Platforms

Programming Languages (PL)
Object-Oriented Programming
Functional Programming
Event-Driven and Reactive Programming
Basic Type Systems

Software Development Fundamentals (SDF)
Algorithms and Design
Fundamental Programming Concepts
Fundamental Data Structures
Development Methods

Software Engineering (SE)
Software Processes
Software Project Management
Tools and Environments
Requirements Engineering
Software Design
Software Construction
Software Verification and Validation

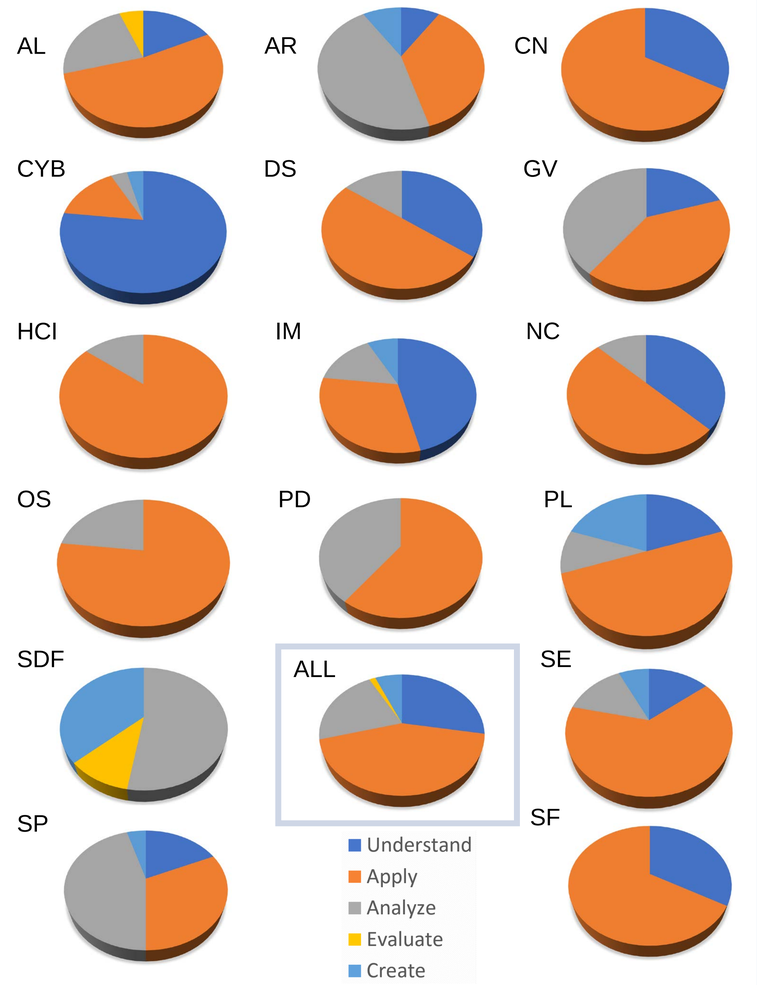
Systems Fundamentals (SF)
Computational Paradigms
Cross-Layer Communications
Parallelism

Social Issues & Professional Practice (SP)
Social Context
Analytical Tools
Professional Ethics
Intellectual Property
Privacy and Civil Liberties
Professional Communication
Sustainability
Security Policies, Laws and Computer Crime

Selected Cyber LOs

Learning Outcome	Emerging	Developed	Highly Developed
[AL-03]. Investigate the use of random/pseudo random number generation in cybersecurity applications. *[Applying]*	Describe the use of random numbers in cybersecurity applications. *[Understanding]*	Investigate the use of random/pseudo random number generation in cybersecurity applications. *[Applying]*	Analyze the use of random/pseudo random number generation in a range of cybersecurity applications. *[Analyzing]*
[AR-03]. Illustrate how fixed-length number representations could affect accuracy and precision, causing vulnerabilities *[Applying]*	Explain how fixed-length number representations could affect accuracy and precision, causing vulnerabilities. *[Understanding]*	Illustrate how fixed-length number representations could affect accuracy and precision, causing vulnerabilities *[Applying]*	Examine how fixed-length number representations could affect accuracy and precision, causing vulnerabilities. *[Analyzing]*
[GV-05]. Perform information hiding through steganography in images, messages, videos, or other media files. *[Applying]*	Demonstrate information hiding through steganography. *[Understanding]*	Perform information hiding through steganography in images, messages, videos, or other media files. *[Applying]*	Choose appropriate steganography technique to conceal information. *[Evaluating]*
[HCI-04]. Investigate the issues of trust in HCI, including examples of both high and low trust systems. *[Applying]*	Demonstrate design elements that make a human-computer interface trustworthy. *[Understanding]*	Investigate the issues of trust in HCI, including examples of both high and low trust systems. *[Applying]*	Critique interface designs between high trust and low trust. *[Evaluating]*
[IM-04]. Describe proven techniques to secure data and information. *[Understanding]*	List proven techniques used to secure data and information. *[Remembering]*	Describe proven techniques to secure data and information. *[Understanding]*	Implement specific proven techniques to secure data and information. *[Applying]*
[NC-06]. Describe security concerns in designing applications for use over wireless networks. *[Understanding]*	Recognize security concerns in designing applications for use over wireless networks.*[Remembering]*	Describe security concerns in designing applications for use over wireless networks. *[Understanding]*	Illustrate security concerns in designing applications for use over wireless networks. *[Applying]*
[OS-11]. Use mechanisms available in an operating system to control access to resources. *[Applying]*	Summarize the mechanisms available in an operating system to control access to resources. *[Understanding]*	Use mechanisms available in an operating system to control access to resources. *[Applying]*	Test the mechanisms available in an operating system to control access to resources. *[Evaluating]*
[PD-04]. Implement mutual exclusion as a way to avoid a given race conditions that could cause security vulnerabilities.*[Applying]*	Explain mutual exclusion as a way to avoid race conditions. *[Understanding]*	Implement mutual exclusion as a way to avoid a given race conditions that could cause security vulnerabilities. *[Applying]*	Categorize critical and noncritical race conditions. *[Analyzing]*
[PL-03]. Use access and visibility modifiers to secure class data and methods. *[Applying]*	Describe access modifiers to secure class data such as private and protected. *[Understanding]*	Use access and visibility modifiers to secure class data and methods. *[Applying]*	Analyze the security effect of using access and visibility modifiers in code. *[Analyzing]*
[SDF-06]. Create programs which use defensive programming techniques, including input validation, type checking, and protection against buffer overflow. *[Creating]*	Investigate defensive programming techniques. *[Applying]*	Create programs which use defensive programming techniques, including input validation, type checking, and protection against buffer overflow.*[Creating]*	Create complex programs which use defensive programming techniques, including input validation, type checking, and protection against buffer overflow. *[Creating]*

Distribution of Bloom's Levels



For More Information

http://ccecc.acm.org/guidance/computer-science-2017

ACM CCECC **Facebook** **Twitter**