Computer Science Program Objectives

AL/Basic Analysis

An ability to demonstrate competency in basic analysis

- **Learning Outcomes**
  - AL-01 Analyze best, average, and worst-case behaviors of an algorithm. [Analyzing]
  - AL-02 Estimate time and space complexities for a given algorithm using Big-O notation. [Evaluating]
  - AL-03 Contrast standard complexity classes. [Analyzing]
  - AL-04 Analyze the performances of an algorithm with various input sizes. [Analyzing]

AL/Algorithmic Strategies

An ability to demonstrate competency in algorithmic strategies

- **Learning Outcomes**
  - AL-05 Apply an appropriate algorithmic approach to a given problem. [Applying]
  - AL-06 Investigate the use of random/pseudo random number generation in cybersecurity applications. [Applying]

AL/Fundamental Data Structures and Algorithms

An ability to demonstrate competency in fundamental data structures and algorithms

- **Learning Outcomes**
  - AL-07 Implement basic numerical algorithms. [Applying]
  - AL-08 Implement common search algorithms, including linear and binary searches. [Applying]
  - AL-09 Implement common sorting algorithms, including iterative, quadratic, and recursive. [Applying]
  - AL-10 Implement hash tables, including collision avoidance and resolution. [Applying]
  - AL-11 Explain the runtime and memory efficiency of principal sorting, searching, and hashing functions. [Understanding]
  - AL-12 Investigate factors other than computational efficiency that influence the choice of algorithms. [Applying]
  - AL-13 Compare various data structures for a given problem. [Analyzing]
  - AL-14 Investigate security vulnerabilities in various data structures. [Applying]
AL/Basic Automata, Computability, and Complexity

An ability to demonstrate competency in basic automata, computability, and complexity

- **Learning Outcomes**
  - AL-15 Write a regular expression to match a pattern. [Applying]
  - AL-16 Describe the concept of finite state machines. [Understanding]
  - AL-17 Explain why the halting problem has no algorithmic solution. [Understanding]

AR/Digital Logic and Digital Systems

An ability to demonstrate competency in digital logic and digital systems

- **Learning Outcomes**
  - AR-01 Diagram the digital components of computing architecture. [Applying]

AR/Machine Level Representation of Data

An ability to demonstrate competency in machine level representation of data

- **Learning Outcomes**
  - AR-02 Analyze alternative formats to represent numerical data. [Analyzing]
  - AR-03 Illustrate how fixed-length number representations could affect accuracy and precision, causing vulnerabilities. [Applying]
  - AR-04 Examine the internal representation of non-numeric data. [Analyzing]
  - AR-05 Compare different methods for converting numerical data from one format to another. [Analyzing]

AR/Assembly Level Machine Organization

An ability to demonstrate competency in assembly level machine organization

- **Learning Outcomes**
  - AR-06 Decompose the organization and major functional units of the classical von Neumann machine. [Analyzing]
  - AR-07 Diagram how high-level language patterns map to assembly/machine language, including subroutine calls. [Applying]
  - AR-08 Create simple assembly language program segments. [Creating]
  - AR-09 Demonstrate the basic concepts of interrupts and I/O operations. [Understanding]

AR/Memory System Organization and Architecture

An ability to demonstrate competency in memory system organization and architecture

- **Learning Outcomes**
AR-10 Compare the cost and performance of different types of memory technology. [Analyzing]
AR-11 Calculate the effect of memory latency on execution time across the memory hierarchy. [Applying]

CN/Introduction to Modeling and Simulation

An ability to demonstrate competency in introduction to modeling and simulation

- **Learning Outcomes**
  - CN-01 Illustrate the concepts of modeling and abstraction with respect to problem solving. [Applying]
  - CN-02 Illustrate the relationship between modeling and simulation. [Applying]
  - CN-03 Exemplify different types of simulations. [Understanding]

CYB/Foundational Concepts in Security

An ability to demonstrate competency in foundational concepts in security

- **Learning Outcomes**
  - CYB-01 Describe security as a continuous process of tradeoffs, balancing between protection mechanisms and availability. [Understanding]
  - CYB-02 Illustrate through examples the concepts of risk, threats, vulnerabilities, attack vectors, and exploits, noting there is no such thing as perfect security. [Applying]
  - CYB-03 Investigate various countermeasures and security controls to minimize risk and exposure. [Applying]
  - CYB-04 Analyze the tradeoffs of balancing key security properties, including Confidentiality, Integrity, and Availability (CIA). [Analyzing]
  - CYB-05 Explain the concepts of trust and trustworthiness related to cybersecurity. [Understanding]
  - CYB-06 Examine ethical issues related to cybersecurity. [Analyzing]
  - CYB-07 Illustrate various ways to minimize privacy risks and maximize anonymity. [Applying]
  - CYB-08 Apply security principles and practices in a dynamic environment. [Applying]
  - CYB-09 Illustrate through examples the key role risk management frameworks play in identifying, assessing, prioritizing, and controlling risks to organizational assets. [Applying]
  - CYB-10 Illustrate with examples the goals of end-to-end data security. [Applying]

CYB/Principles of Secure Design

An ability to demonstrate competency in principles of secure design

- **Learning Outcomes**
  - CYB-11 Use the principles of secure design. [Applying]
  - CYB-12 Illustrate the security implications of relying on open design vs the secrecy of design. [Applying]
  - CYB-13 Discuss the benefits and limitations of designing multiple layers of defenses. [Understanding]
  - CYB-14 Analyze the tradeoffs associated with designing security into a product. [Analyzing]

CYB/Defensive Programming
An ability to demonstrate competency in defensive programming

- **Learning Outcomes**
  CYB-15 Construct input validation and data sanitization in applications, considering adversarial control of the input channel. [Creating] See also SDF-06.
  CYB-16 Explain the tradeoffs of developing a program in a type-safe language. [Understanding] See also PL-10.
  CYB-17 Implement programs that properly handle exceptions and error conditions. [Applying] See also SDF-12.
  CYB-18 Examine the need to update software to fix security vulnerabilities. [Analyzing]

**CYB/Threats and Attacks**

An ability to demonstrate competency in threats and attacks

- **Learning Outcomes**
  CYB-19 Examine likely attack types against standalone and networked systems. [Analyzing]
  CYB-20 Illustrate the key principles of social engineering, including membership and trust. [Applying]
  See also SP-04.

**CYB/Cryptography**

An ability to demonstrate competency in cryptography

- **Learning Outcomes**
  CYB-21 Describe key terms in cryptology, including cryptography, cryptanalysis, cipher, cryptographic algorithm, and public key infrastructure. [Understanding]
  CYB-22 Use a variety of ciphers to encrypt plaintext into ciphertext. [Applying]
  CYB-23 Apply cryptographic hash functions for authentication and data integrity. [Applying]
  CYB-24 Contrast symmetric and asymmetric encryption in relation to securing electronic communications and transactions. [Analyzing]

**CYB/Web Security**

An ability to demonstrate competency in web security

- **Learning Outcomes**
  CYB-25 Explain browser and web security model concepts including same-origin policy, web sessions, and secure communication channels. [Understanding]

**DS/Sets, Relations, and Functions**

An ability to demonstrate competency in sets, relations, and functions.
• **Learning Outcomes**
  DS-01 Explain with examples the basic terminology of functions, relations, and sets. [Understanding]
  DS-02 Perform the operations associated with sets, functions, and relations. [Applying]
  DS-03 Compare practical examples to the appropriate set, function, or relation model, and interpret the
  associated operations and terminology in context. [Analyzing]

**DS/Basic Logic**

An ability to demonstrate competency in basic logic

• **Learning Outcomes**
  DS-04 Convert logical statements from informal language to propositional and predicate logic
  expressions. [Understanding]
  DS-05 Apply formal logic proofs and/or informal, but rigorous, logical reasoning to real problems such as
  predicting the behavior of software or solving problems such as puzzles. [Applying]
  DS-06 Use the rules of inference to construct proofs in propositional and predicate logic. [Applying]
  DS-07 Describe how symbolic logic can be used to model real-life situations or computer applications.
  [Understanding]
  DS-08 Apply formal methods of symbolic propositional and predicate logic, such as calculating validity
  of formulae and computing normal forms. [Applying]
  DS-09 Describe the strengths and limitations of propositional and predicate logic. [Understanding]

**DS/Proof Techniques**

An ability to demonstrate competency in proof techniques

• **Learning Outcomes**
  DS-10 Outline the basic structure of each proof technique, including direct proof, proof by contradiction,
  and induction. [Analyzing]
  DS-11 Apply each of the proof techniques (direct proof, proof by contradiction, and induction) correctly
  in the construction of a sound argument. [Applying]
  DS-12 Deduce the best type of proof for a given problem. [Analyzing]
  DS-13 Explain the parallels between ideas of mathematical and/or structural induction to recursion and
  recursively defined structures. [Understanding]
  DS-14 Explain the relationship between weak and strong induction and give examples of the appropriate
  use of each. [Understanding]

**DS/Basics of Counting**

An ability to demonstrate competency in basic of counting

• **Learning Outcomes**
  DS-15 Apply counting arguments, including sum and product rules, inclusion-exclusion principle and
  arithmetic/geometric progressions. [Applying]
  DS-16 Apply the pigeonhole principle in the context of a formal proof. [Applying]
  DS-17 Calculate permutations and combinations of a set, and interpret the meaning in the context of the
particular application. [Applying]
DS-18 Compare real-world applications appropriate to counting formalisms. [Analyzing]
DS-19 Solve a variety of basic recurrence relations. [Applying]
DS-20 Analyze a problem to determine underlying recurrence relations. [Analyzing]
DS-21 Perform computations involving modular arithmetic. [Applying]

**DS/Graphs and Trees**

An ability to demonstrate competency in graphs and trees

- **Learning Outcomes**
  DS-22 Illustrate the basic terminology of graph theory including properties and special cases for each type of graph/tree. [Applying]
  DS-23 Demonstrate different traversal methods for trees and graphs, including pre-, post-, and in-order traversal of trees. [Understanding]
  DS-24 Solve a variety of real-world problems in computer science using appropriate forms of graphs and trees, such as representing a network topology or the organization of a hierarchical file system. [Applying]
  DS-25 Implement and use balanced trees and B-trees. [Applying]
  DS-26 Implement graph algorithms. [Applying]
  DS-27 Demonstrate how concepts from graphs and trees appear in data structures, algorithms, proof techniques (structural induction), and counting. [Understanding]
  DS-28 Describe binary search trees and AVL trees. [Understanding]
  DS-29 Explain complexity in the ideal and in the worst-case scenario for both implementations. [Understanding]

**DS/Discrete Probability**

An ability to demonstrate competency in discrete probability

- **Learning Outcomes**
  DS-30 Calculate probabilities of events and expectations of random variables for elementary problems. [Applying]
  DS-31 Differentiate between dependent and independent events. [Understanding]
  DS-32 Explain the significance of binomial distribution in probabilities. [Understanding]
  DS-33 Apply Bayes Theorem to determine conditional probabilities in a problem. [Applying]
  DS-34 Apply the tools of probability to solve problems. [Applying]

**GV/Fundamental Concepts**

An ability to demonstrate competency in fundamental concepts

- **Learning Outcomes**
  GV-01 Compare transformation and changes in dimension and coordinate systems for 2D and 3D design. [Analyzing]
  GV-02 Demonstrate common uses of digital presentation to human senses. [Understanding]
  GV-03 Illustrate color models and their use in computer graphics. [Applying]
GV-04 Analyze image types according to output choices. [Analyzing]
GV-05 Perform information hiding through steganography in images, messages, videos, or other media files. [Applying]

**HCl/Foundations**

An ability to demonstrate competency in fundations

- **Learning Outcomes**
  - HCI-01 Analyze the importance of human-centered software. [Analyzing]
  - HCI-02 Implement a simple usability test for an existing software application. [Applying]
  - HCI-03 Examine the issues of trust in HCI, including examples of both high and low trust systems. [Analyzing]

**HCl/Designing Interaction**

An ability to demonstrate competency in designing Interaction

- **Learning Outcomes**
  - HCI-04 Write a simple application that uses a modern graphical user interface. [Applying]
  - HCI-05 Use at least one national or international user interface design standard in a simple application. [Applying] See also SP-02.
  - HCI-06 Analyze the interaction between a security mechanism and its usability. [Analyzing]

**IM/Information Management Concepts**

An ability to demonstrate competency in information management concepts

- **Learning Outcomes**
  - IM-01 Contrast the difference between information and data. [Analyzing]
  - IM-02 Describe the advantages and disadvantages of central organizational control over data. [Understanding]
  - IM-03 Investigate contingency planning with respect to business continuity, disaster recovery and incident response. [Applying]
  - IM-04 Describe proven techniques to secure data and information. [Understanding]
  - IM-05 Describe approaches to scale up information systems. [Understanding]

**IM/Database Systems**

An ability to demonstrate competency in database systems

- **Learning Outcomes**
  - IM-06 Explain the characteristics that distinguish the database approach from the approach of programming with data files. [Understanding]
  - IM-07 Diagram the components of a database system and give examples of their use. [Applying]
IM-08 Explain the concept of data independence and its importance in a database system. [Understanding]
IM-09 Formulate queries in SQL or a similar query language to elicit information from a database. [Creating]
IM-10 Investigate vulnerabilities and failure scenarios in database systems. [Applying]

**IM/Data Modeling**

An ability to demonstrate competency in data modeling

- **Learning Outcomes**
  IM-11 Contrast appropriate data models, including internal structures for different data types. [Analyzing]
  IM-12 Diagram a relational data model for a given scenario that addresses security and privacy concerns. [Applying]
  IM-13 Describe the differences among relational data models and other models. [Understanding]

**NC/Introduction**

An ability to demonstrate competency in introduction

- **Learning Outcomes**
  NC-01 Diagram the basic structure of the Internet. [Applying]
  NC-02 Describe the layered structure of a typical networked architecture, including routing and switching. [Understanding]
  NC-03 Diagram the layers of the OSI model, including associated protocols. [Applying]

**NC/Networked Applications**

An ability to demonstrate competency in networked applications

- **Learning Outcomes**
  NC-04 Categorize the principles used for naming schemes and resource location. [Analyzing]
  NC-05 Implement a simple distributed network application. [Applying]
  NC-06 Describe security concerns in designing applications for use over wireless networks. [Understanding]
  NC-07 Illustrate secure connectivity among networked applications. [Applying]
  NC-08 Explain the advantages and disadvantages of using virtualized infrastructure in cloud computing. [Understanding] See also PD-05

**OS/Overview of Operating Systems**

An ability to demonstrate competency in overview of operating systems

- **Learning Outcomes**
  OS-01 Examine major objectives, functions, features, and concepts of modern operating systems. [Analyzing]
OS/Operating System Principles

An ability to demonstrate competency in operating system principles

- **Learning Outcomes**
  - OS-04 Diagram the interaction of an Application Programming Interface (API) with an operating system.
  - OS-05 Illustrate how computing resources are used by applications and managed by the operating system.
  - OS-06 Manipulate a device list or driver I/O queue.

OS/Concurrency

An ability to demonstrate competency in concurrency

- **Learning Outcomes**
  - OS-07 Investigate the need for concurrency within an operating system.

OS/Memory Management

An ability to demonstrate competency in memory management

- **Learning Outcomes**
  - OS-08 Illustrate the principles of memory management.
  - OS-09 Illustrate the concepts of virtual memory, including paging, thrashing, and partitioning.

OS/Security and Protection

An ability to demonstrate competency in security and protection

- **Learning Outcomes**
  - OS-10 Investigate the features and limitations of an operating system used to provide protection and security.
  - OS-11 Use mechanisms available in an operating system to control access to resources.

OS/Virtual Machines

An ability to demonstrate competency in virtual machines

- **Learning Outcomes**
  - OS-12 Analyze the concept of virtualization with respect to hardware and software.
OS-13 Diagram the physical hardware devices and the virtual devices maintained by an operating system. [Applying]

PD/Parallelism Fundamentals

An ability to demonstrate competency in parallelism fundamentals

- **Learning Outcomes**
  PD-01 Analyze the goals of parallelism and concurrency. [Analyzing] See also OS-07 and SF-08.
  PD-02 Implement various programming constructs for synchronization. [Applying]
  PD-03 Contrast low-level data races with higher level races. [Analyzing]

PD/Communication and Coordination

An ability to demonstrate competency in communication and coordination

- **Learning Outcomes**
  PD-04 Implement mutual exclusion in order to avoid race conditions that could cause security vulnerabilities. [Applying]

PD/Cloud Computing

An ability to demonstrate competency in cloud computing

- **Learning Outcomes**
  PD-05 Investigate the challenges and concerns related to security and privacy in Cloud computing. [Applying] See also NC-08.

PL/Object-Oriented Programming

An ability to demonstrate competency in object-oriented programming

- **Learning Outcomes**
  PL-01 Design a simple class hierarchy using superclasses, subclasses, and abstract classes. [Creating]
  PL-02 Diagram control flow in a program using dynamic dispatch. [Applying]
  PL-03 Use access and visibility modifiers to secure class data and methods. [Applying]
  PL-04 Implement in code OOP constructs, including encapsulation, abstraction, inheritance, and polymorphism. [Applying]

PL/Functional Programming

An ability to demonstrate competency in functional programming

- **Learning Outcomes**
PL-05 Implement algorithms which utilize immutable and mutable variables. [Applying]
PL-06 Contrast functional and object-oriented programming paradigms. [Analyzing]

**PL/Event-Driven and Reactive Programming**

An ability to demonstrate competency in event-driven and reactive programming

- **Learning Outcomes**
  - PL-07 Create an interactive program using an event-driven style. [Creating]
  - PL-08 Describe potential security vulnerabilities in event-driven GUI applications. [Understanding]

**PL/Basic Type Systems**

An ability to demonstrate competency in basic type systems

- **Learning Outcomes**
  - PL-09 Investigate potential errors detected from both strong-type and weak-type languages. [Applying]
  - PL-10 Explain the security implications of a type-safe language for software development. [Understanding]

**SDF/Algorithms and Design**

An ability to demonstrate competency in algorithms and design

- **Learning Outcomes**
  - SDF-01 Design an algorithm in a programming language to solve a simple problem. [Creating]
  - SDF-02 Use the techniques of decomposition to modularize a program. [Applying]
  - SDF-03 Compare multiple algorithms for a given problem. [Analyzing]

**SDF/Fundamental Programming Concepts**

An ability to demonstrate competency in fundamental programming concepts

- **Learning Outcomes**
  - SDF-04 Create simple programs that use abstract data types (ADTs). [Creating]
  - SDF-05 Investigate potential vulnerabilities in provided programming code. [Applying] See also AL-16, CYB-28, PL-10, SDF-12.
  - SDF-06 Create programs which use defensive programming techniques, including input validation, type checking, and protection against buffer overflow. [Creating] See also CYB-15.
  - SDF-07 Create code in a programming language that includes primitive data types, references, variables, expressions, assignments, I/O, control structures, and functions. [Creating]
  - SDF-08 Create a simple program that uses persistence to save data across multiple executions. [Creating]
  - SDF-09 Create a simple program that uses recursion. [Creating]
SDF/Fundamental Data Structures

An ability to demonstrate competency in fundamental data structures

- **Learning Outcomes**
  SDF-10 Create simple programs that include each of the following data structures: lists, stacks, queues, hash tables, graphs and trees. [Creating]
  SDF-11 Compare the efficiency of basic operations across various data structures. [Analyzing]

SDF/Development Methods

An ability to demonstrate competency in development methods

- **Learning Outcomes**
  SDF-12 Investigate common coding errors that introduce security vulnerabilities. [Applying]
  SDF-13 Implement refactoring within given program components. [Applying]
  SDF-14 Analyze programming code that utilizes preconditions, postconditions, and invariants. [Analyzing]
  SDF-15 Apply a variety of strategies to test and debug programs. [Applying]
  SDF-16 Use an integrated development environment (IDE) to create, execute, test, and debug secure programs. [Applying]
  SDF-17 Use standard libraries for a given programming language. [Applying]
  SDF-18 Apply consistent documentation and program style standards. [Applying]
  SDF-19 Carry out a code review on a program component using a provided security checklist. [Applying]

SE/Software Processes

An ability to demonstrate competency in software processes

- **Learning Outcomes**
  SE-01 Diagram how software interacts with various systems, including information management, embedded, process control, and communications systems. [Applying]
  SE-02 Compare the features of various process models. [Analyzing]
  SE-03 Diagram the phases of the secure software development lifecycle (SecSDLC). [Applying]

SE/Software Project Management

An ability to demonstrate competency in software project management

- **Learning Outcomes**
  SE-04 Illustrate common behaviors that contribute to the effective functioning of a team. [Applying]
  SE-05 Investigate the risks in using third-party applications, software tools, and libraries. [Applying]
SE/Tools and Environments

An ability to demonstrate competency in tools and environments

- **Learning Outcomes**
  - SE-06 Use a set of development tools for software systems. [Applying]

SE/Requirements Engineering

An ability to demonstrate competency in requirements engineering

- **Learning Outcomes**
  - SE-07 Implement the requirements for a secure software system. [Applying]

SE/Software Design

An ability to demonstrate competency in software design

- **Learning Outcomes**
  - SE-08 Illustrate principles of secure software design [Applying]
  - SE-09 Analyze an existing software design to improve its security. [Analyzing]
  - SE-10 Describe the cost and tradeoffs associated with designing security into software. [Understanding]

SE/Software Construction

An ability to demonstrate competency in software construction

- **Learning Outcomes**
  - SE-11 Implement a small software project that uses a defined coding standard. [Applying]

SE/Software Verification and Validation

An ability to demonstrate competency in software verification and validation

- **Learning Outcomes**
  - SE-12 Differentiate between program validation and verification. [Understanding]
  - SE-13 Implement in code different types of testing, including security, unit testing, system testing, integration testing, and interface usability. [Applying]
  - SE-14 Design a test plan that validates software security. [Creating]

SF/Computational Paradigms
An ability to demonstrate competency in computational paradigms

- **Learning Outcomes**
  SF-01 Illustrate the basic building blocks of computers and their role in the historical development of computer architecture. [Applying]
  SF-02 Discuss the differences between single thread and multiple thread, as well as single server and multiple server models. [Understanding]
  SF-03 Investigate security implications related to emerging computational paradigms. [Applying]

**SF/Cross-Layer Communications**

An ability to demonstrate competency in cross-layer communications

- **Learning Outcomes**
  SF-04 Describe how computing systems are constructed of layers upon layers. [Understanding]
  SF-05 Implement a program using methods of layering. [Applying]
  SF-06 Investigate defects in a layered program using tools for program tracing, single stepping, and debugging. [Applying]

**SF/Parallelism**

An ability to demonstrate competency in parallelism

- **Learning Outcomes**
  SF-07 Illustrate the performance of simple sequential and parallel versions of the same program with different problem sizes. [Applying]
  SF-08 Summarize the differences among the concepts of instruction parallelism, data parallelism, thread parallelism/multitasking, and task/request parallelism. [Understanding] See also PD-01.
  SF-09 Investigate other uses of parallelism, including reliability and redundancy of execution. [Applying]

**SP/Social Context**

An ability to demonstrate competency in social context

- **Learning Outcomes**
  SP-01 Investigate both positive and negative ways in which computing technology impacts information exchange and social interaction. [Applying]
  SP-02 Examine developers’ assumptions and values embedded in hardware and software design, especially with respect to underrepresented groups and diverse populations. [Analyzing] See also HCI-06.
  SP-03 Analyze the impact of diversity on the computing profession. [Analyzing]
  SP-04 Investigate social engineering attacks and the types of bad actors who might perform them. [Applying]
SP/Analytical Tools

An ability to demonstrate competency in analytical tools

- **Learning Outcomes**
  SP-05 Contrast stakeholder positions in a given scenario. [Analyzing]
  SP-06 Analyze social tradeoffs in technical decisions. [Analyzing]

SP/Professional Ethics

An ability to demonstrate competency in professional ethics

- **Learning Outcomes**
  SP-07 Examine various ethics scenarios in computing. [Analyzing]
  SP-08 Support the ethical responsibility of ensuring software correctness, reliability, and safety. [Evaluating]
  SP-09 Compare professional codes of conduct from the ACM, IEEE Computer Society, and other organizations. [Analyzing]

SP/Intellectual Property

An ability to demonstrate competency in intellectual property

- **Learning Outcomes**
  SP-10 Differentiate the terms among intellectual property, fair-use, copyright, patent, trademark, and plagiarism. [Understanding]
  SP-11 Discuss the rationale for legal protection of intellectual property. [Understanding]

SP/Privacy and Civil Liberties

An ability to demonstrate competency in privacy and civil liberties

- **Learning Outcomes**
  SP-12 Outline the need for legal protection of personal privacy. [Analyzing]
  SP-13 Investigate threats to privacy rights in personally identifiable information (PII). [Applying]
  SP-14 Illustrate the role of data collection in the implementation of pervasive surveillance systems. [Applying]
  SP-15 Analyze technological solutions to privacy concerns. [Analyzing]

SP/Professional Communication

An ability to demonstrate competency in professional communication

- **Learning Outcomes**
SP-16 Use effective oral, written, electronic, and visual communication techniques with stakeholders. [Applying]
SP-17 Interpret the impact of both verbal and nonverbal cues during communication among team members. [Understanding]
SP-18 Develop technical artifacts. [Creating]

SP/Sustainability

An ability to demonstrate competency in sustainability

- Learning Outcomes
  SP-19 Analyze case studies related to sustainable computing efforts. [Analyzing]

SP/Security Policies, Laws and Computer Crime

An ability to demonstrate competency in security policies, laws and computer crime

- Learning Outcomes
  SP-20 Investigate laws applicable to computer crimes. [Applying]
  SP-21 Examine the motivation and ramifications of cyber terrorism and criminal hacking. [Analyzing]
  SP-22 Write a company-wide security policy. [Applying]

Association for Computing Machinery

- Login
- Acceptable Use
- Privacy Policy

Copyright © 2019 ACM CCECC