**Computer Science Metrics**

AL-01 Analyze best, average, and worst-case behaviors of an algorithm. [Analyzing]

Emerging Standard
Illustrate best, average, and worst-case behaviors of an algorithm. [Applying]
Developed Standard
Analyze best, average, and worst-case behaviors of an algorithm. [Analyzing]
Highly Developed Standard
Evaluate best, average, and worst-case behaviors of an algorithm. [Evaluating]

AL-02 Estimate time and space complexities for a given algorithm using Big-O notation. [Evaluating]

Emerging Standard
Distinguish between time and space complexities for a given algorithm. [Analyzing]
Developed Standard
Estimate time and space complexities for a given algorithm using Big-O notation. [Evaluating]
Highly Developed Standard
Critique time and space complexities of several algorithms using Big-O and other notations. [Evaluating]

AL-03 Contrast standard complexity classes. [Analyzing]

Emerging Standard
Illustrate a few of the standard complexity classes. [Applying]
Developed Standard
Contrast standard complexity classes, such as logarithmic, linear, quadratic, and exponential. [Analyzing]
Highly Developed Standard
Judge standard complexity classes as either "efficient" or "inefficient" algorithms. [Evaluating]

AL-04 Analyze the performances of an algorithm with various input sizes. [Analyzing]

Emerging Standard
Discuss the performances of an algorithm with various input sizes. [Understanding]
Developed Standard
Compare the performances of an algorithm with various input sizes. [Analyzing]
Highly Developed Standard
Assess the performances of an algorithm with various input sizes [Evaluating]

**AL-05 Apply an appropriate algorithmic approach to a given problem. [Applying]**

Emerging Standard  
Demonstrate an algorithmic approach to a given problem. [Understanding]  
Developed Standard  
Apply an appropriate algorithmic approach to a given problem, such as brute-force, greedy, recursive, divide-and-conquer, and dynamic programming. [Applying]  
Highly Developed Standard  
Analyze the tradeoffs of various algorithmic approaches to a given problem. [Analyzing]

**AL-06 Investigate the use of random/pseudo random number generation in cybersecurity applications. [Applying]**

Emerging Standard  
Describe the use of random numbers in cybersecurity applications. [Understanding]  
Developed Standard  
Investigate the use of random/pseudo random number generation in cybersecurity applications, such as password generation and data encryption. [Applying]  
Highly Developed Standard  
Analyze the use of random/pseudo random number generation in a range of cybersecurity applications. [Analyzing]

**AL-07 Implement basic numerical algorithms. [Applying]**

Emerging Standard  
Describe the use of basic numerical algorithms. [Understanding]  
Developed Standard  
Implement basic numerical algorithms, such as min, max, and mode. [Applying]  
Highly Developed Standard  
Develop complex numerical algorithms. [Creating]

**AL-08 Implement common search algorithms, including linear and binary searches. [Applying]**

Emerging Standard  
Exemplify common search algorithms, including linear and binary searches. [Understanding]  
Developed Standard  
Implement common search algorithms, including linear and binary searches. [Applying]  
Highly Developed Standard  
Compare the efficiency of common search algorithms, including linear and binary searches. [Analyzing]
AL-09 Implement common sorting algorithms, including iterative, quadratic, and recursive. [Applying]

Emerging Standard
Exemplify common sorting algorithms, including iterative, quadratic, and recursive. [Understanding]
Developed Standard
Implement common sorting algorithms, including iterative, quadratic, and recursive. [Applying]
Highly Developed Standard
Compare the efficiency of common sorting algorithms, including iterative, quadratic, and recursive. [Analyzing]

AL-10 Implement hash tables, including collision avoidance and resolution. [Applying]

Emerging Standard
Explain the general idea of a hash table. [Understanding]
Developed Standard
Implement hash tables, including collision avoidance and resolution. [Applying]
Highly Developed Standard
Compare common collision resolution techniques for hash tables. [Analyzing]

AL-11 Explain the runtime and memory efficiency of principal sorting, searching, and hashing functions. [Understanding]

Emerging Standard
Summarize the runtime and memory efficiency of either a sorting, a searching, or a hashing function. [Understanding]
Developed Standard
Explain the runtime and memory efficiency of principal sorting, searching, and hashing functions. [Understanding]
Highly Developed Standard
Analyze the runtime and memory efficiency of principal sorting, searching, and hashing functions. [Analyzing]

AL-12 Investigate factors other than computational efficiency that influence the choice of algorithms. [Applying]

Emerging Standard
Describe factors other than computational efficiency that ought to be considered when choosing an algorithm. [Understanding]
Developed Standard
Investigate factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the size and patterns of input data. [Applying]
Highly Developed Standard
Critique factors other than computational efficiency that influence the choice of algorithms. [Evaluating]
AL-13 Compare various data structures for a given problem. [Analyzing]

Emerging Standard
Investigate a few data structures for a given problem. [Applying]
Developed Standard
Compare various data structures for a given problem, such as array, list, set, map, stack, queue, hash table, tree, and graph. [Analyzing]
Highly Developed Standard
Justify a choice of data structure for a given problem. [Evaluating]

AL-14 Investigate security vulnerabilities in various data structures. [Applying]

Emerging Standard
Summarize security vulnerabilities in various data structures [Understanding]
Developed Standard
Investigate security vulnerabilities in various data structures, such as out-of-bounds arrays and buffer overflows. [Applying]
Highly Developed Standard
Analyze security vulnerabilities in various data structures. [Analyzing]

AL-15 Write a regular expression to match a pattern. [Applying]

Emerging Standard
Explain the use of regular expressions in pattern matching. [Understanding]
Developed Standard
Write a regular expression to match a pattern. [Applying]
Highly Developed Standard
Write a regular expression to perform complex pattern matching. [Applying]

AL-16 Describe the concept of finite state machines. [Understanding]

Emerging Standard
Recognize a finite state machine. [Remembering]
Developed Standard
Describe the concept of finite state machines. [Understanding]
Highly Developed Standard
Diagram a finite state machine. [Applying]

AL-17 Explain why the halting problem has no algorithmic solution. [Understanding]

Emerging Standard
Recognize that some problems have no algorithmic solution. [Remembering]
Explain why the halting problem has no algorithmic solution. [Understanding]

Illustrate a proof of the halting problem. [Applying]

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

Discuss some of the digital components of computing architecture. [Understanding]

Diagram the digital components of computing architecture, such as logic gates, registers, and memory. [Applying]

Evaluate the digital component design of a computing architecture for accuracy. [Evaluating]

**AR-02 Analyze alternative formats to represent numerical data.** [Analyzing]

Explain reasons for using alternative formats to represent numerical data. [Understanding]

Analyze alternative formats to represent numerical data. [Analyzing]

Choose optimal formats to represent a given set of numerical data. [Evaluating]

**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]

**AR-04 Examine the internal representation of non-numeric data.** [Analyzing]

Illustrate internal representation of non-numeric data. [Applying]

Examine the internal representation of non-numeric data, such as characters, strings, records, and arrays. [Analyzing]

Highly Developed Standard
Estimate memory requirements for non-numeric data, such as characters, strings, records, and arrays. [Evaluating]

**AR-05 Compare different methods for converting numerical data from one format to another.** [Analyzing]

Emerging Standard
Convert numerical data from one format to another. [Understanding]

Developed Standard
Compare different methods for converting numerical data from one format to another, such as converting negative integers into sign-magnitude and two’s-complement representations. [Analyzing]

Highly Developed Standard
Evaluate different methods for converting numerical data from one format to another. [Evaluating]

**AR-06 Decompose the organization and major functional units of the classical von Neumann machine.** [Analyzing]

Emerging Standard
Diagram the organization of the classical von Neumann machine and its major functional units. [Applying]

Developed Standard
Decompose the organization and major functional units of the classical von Neumann machine. [Analyzing]

Highly Developed Standard
Assess the organization of the classical von Neumann machine and its major functional units. [Evaluating]

**AR-07 Diagram how high-level language patterns map to assembly/machine language, including subroutine calls.** [Applying]

Emerging Standard
Summarize how high-level language patterns map to assembly/machine language, including subroutine calls. [Understanding]

Developed Standard
Diagram how high-level language patterns map to assembly/machine language, including subroutine calls. [Applying]

Highly Developed Standard
Examine how high-level language patterns map to assembly/machine language, including subroutine calls. [Analyzing]

**AR-08 Create simple assembly language program segments.** [Creating]

Emerging Standard
Implement simple assembly language program segments. [Applying]

Developed Standard
Create simple assembly language program segments. [Creating]

Highly Developed Standard
Create complex assembly language program segments. [Creating]
AR-09 Demonstrate the basic concepts of interrupts and I/O operations. [Understanding]

Emerging Standard
List basic concepts of interrupts and I/O operations. [Remembering]
Developed Standard
Demonstrate the basic concepts of interrupts and I/O operations. [Understanding]
Highly Developed Standard
Implement basic concepts of interrupts and I/O operations. [Applying]

AR-10 Compare the cost and performance of different types of memory technology. [Analyzing]

Emerging Standard
Describe different types of memory technology. [Understanding]
Developed Standard
Compare the cost and performance of different types of memory technology, such as SRAM, DRAM, virtual, and cache. [Analyzing]
Highly Developed Standard
Critique the cost and performance of different types of memory technology. [Evaluating]

AR-11 Calculate the effect of memory latency on execution time across the memory hierarchy. [Applying]

Emerging Standard
Explain the effect of memory latency on execution time across the memory hierarchy. [Understanding]
Developed Standard
Calculate the effect of memory latency on execution time across the memory hierarchy. [Applying]
Highly Developed Standard
Examine the effect of memory latency on execution time across the memory hierarchy. [Analyzing]

CN-01 Illustrate the concepts of modeling and abstraction with respect to problem solving. [Applying]

Emerging Standard
Explain the concepts of modeling and abstraction with respect to problem solving. [Understanding]
Developed Standard
Illustrate the concepts of modeling and abstraction with respect to problem solving. [Applying]
Highly Developed Standard
Contrast the concepts of modeling and abstraction with respect to problem solving. [Analyzing]

CN-02 Illustrate the relationship between modeling and simulation. [Applying]

Emerging Standard
Describe the relationships between modeling and simulation. [Understanding]
Developed Standard
Illustrate the relationship between modeling and simulation. [Applying]
Highly Developed Standard
Examine the relationship between modeling and simulation. [Analyzing]

**CN-03 Exemplify different types of simulations. [Understanding]**

Emerging Standard
Identify different types of simulations. [Remembering]

Developed Standard
Exemplify different types of simulations, such as physical simulations, human-guided simulations, and virtual reality. [Understanding]

Highly Developed Standard
Compare different types of simulations. [Analyzing]

**CYB-01 Describe security as a continuous process of tradeoffs, balancing between protection mechanisms and availability. [Understanding]**

Emerging Standard
Recognize the importance of security as a continuous process of tradeoffs. [Remembering]

Developed Standard
Describe security as a continuous process of tradeoffs, balancing between protection mechanisms and availability. [Understanding]

Highly Developed Standard
Illustrate security as a continuous process of tradeoffs, balancing between protection mechanisms and availability. [Applying]

**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]**

Emerging Standard
Differentiate the concepts of risk, threats, vulnerabilities, attack vectors, and exploits. [Understanding]

Developed Standard
Illustrate through examples the concepts of risk, threats, vulnerabilities, attack vectors, and exploits, noting there is no such thing as perfect security. [Applying]

Highly Developed Standard
Compare the concepts of risk, threats, vulnerabilities, attack vectors, and exploits to a given scenario. [Analyzing]

**CYB-03 Investigate various countermeasures and security controls to minimize risk and exposure. [Applying]**

Emerging Standard
Explain the importance of countermeasures and security controls to minimize risk and exposure. [Understanding]

Developed Standard
Investigate various countermeasures and security controls to minimize risk and exposure. [Applying]

Highly Developed Standard
Choose among various countermeasures and security controls to minimize risk and exposure in a given scenario. [Evaluating]

**CYB-04 Analyze the tradeoffs of balancing key security properties, including Confidentiality, Integrity, and Availability (CIA). [Analyzing]**

Emerging Standard
Investigate the tradeoffs of balancing key security properties, including Confidentiality, Integrity, and Availability (CIA). [Applying]

Developed Standard
Analyze the tradeoffs of balancing key security properties, including Confidentiality, Integrity, and Availability (CIA). [Analyzing]

Highly Developed Standard
Evaluate the tradeoffs of balancing key security properties, including Confidentiality, Integrity, and Availability (CIA). [Evaluating]

**CYB-05 Explain the concepts of trust and trustworthiness related to cybersecurity. [Understanding]**

Emerging Standard
Define the concepts of trust and trustworthiness. [Remembering]

Developed Standard
Explain the concepts of trust and trustworthiness related to cybersecurity. [Understanding]

Highly Developed Standard
Diagram trust relationships in a given cybersecurity scenario. [Applying]

**CYB-06 Examine ethical issues related to cybersecurity. [Analyzing]**

Emerging Standard
Exemplify ethical issues in cybersecurity. [Understanding]

Developed Standard
Examine ethical issues related to cybersecurity. [Analyzing]

Highly Developed Standard
Argue ethical issues related to cybersecurity. [Evaluating]

**CYB-07 Illustrate various ways to minimize privacy risks and maximize anonymity. [Applying]**

Emerging Standard
Describe one way to minimize privacy risks and maximize anonymity. [Understanding]

Developed Standard
Illustrate various ways to minimize privacy risks and maximize anonymity. [Applying]

Highly Developed Standard
Analyze ways to minimize privacy risks and maximize anonymity in an always connected, mobile computing environment. [Analyzing]
CYB-08 Apply security principles and practices in a dynamic environment. [Applying]

Emerging Standard
Summarize security principles and practices. [Understanding]

Developed Standard
Apply security principles and practices in a dynamic environment. [Applying]

Highly Developed Standard
Examine security principles and practices in a dynamic environment. [Analyzing]

CYB-09 Illustrate through examples the key role risk management frameworks play in identifying, assessing, prioritizing, and controlling risks to organizational assets. [Applying]

Emerging Standard
Paraphrase the key role risk management frameworks play in identifying, assessing, prioritizing, and controlling risks to organizational assets. [Understanding]

Developed Standard
Illustrate through examples the key role risk management frameworks play in identifying, assessing, prioritizing, and controlling risks to organizational assets. [Applying]

Highly Developed Standard
Analyze a given scenario with a specific risk management framework, such as NIST, to identify, assess, prioritize, and control risks to organizational assets. [Analyzing]

CYB-10 Illustrate with examples the goals of end-to-end data security. [Applying]

Emerging Standard
Explain the goals of end-to-end data security. [Understanding]

Developed Standard
Illustrate with examples the goals of end-to-end data security. [Applying]

Highly Developed Standard
Outline the goals of end-to-end data security. [Analyzing]

CYB-11 Use the principles of secure design. [Applying]

Emerging Standard
Demonstrate some of the principles of secure design as related to cybersecurity. [Understanding]

Developed Standard
Use the principles of secure design, such as least privilege, isolation, fail-safe, and deny-by-default. [Applying]

Highly Developed Standard
Choose appropriate secure design principles for a given cybersecurity scenario. [Evaluating]

CYB-12 Illustrate the security implications of relying on open design vs the secrecy of design. [Applying]
Emerging Standard
Discuss the security implications of relying on open design vs the secrecy of design. [Understanding]
Developed Standard
Illustrate the security implications of relying on open design vs the secrecy of design. [Applying]
Highly Developed Standard
Analyze the security implications of relying on open design vs the secrecy of design. [Analyzing]

CYB-13 Discuss the benefits and limitations of designing multiple layers of defenses. [Understanding]
Emerging Standard
Identify the benefits and limitations of designing multiple layers of defenses. [Remembering]
Developed Standard
Discuss the benefits and limitations of designing multiple layers of defenses. [Understanding]
Highly Developed Standard
Implement multiple layers of defenses for a given scenario [Applying]

CYB-14 Analyze the tradeoffs associated with designing security into a product. [Analyzing]
Emerging Standard
Summarize the tradeoffs associated with designing security into a product. [Understanding]
Developed Standard
Analyze the tradeoffs associated with designing security into a product. [Analyzing]
Highly Developed Standard
Evaluate the tradeoffs associated with designing security into a product. [Evaluating]

CYB-15 Construct input validation and data sanitization in applications, considering adversarial control of the input channel. [Creating] See also SDF-06.
Emerging Standard
Implement simple input validation and data sanitization in applications. [Applying]
Developed Standard
Construct input validation and data sanitization in applications, considering adversarial control of the input channel. [Creating]
Highly Developed Standard
Develop complex input validation and data sanitization in applications, considering adversarial control of the input channel. [Creating]

CYB-16 Explain the tradeoffs of developing a program in a type-safe language. [Understanding] See also PL-10.
Emerging Standard
List some of the tradeoffs of developing a program in a type-safe language. [Remembering]
Developed Standard
Explain the tradeoffs of developing a program in a type-safe language. [Understanding]
Highly Developed Standard
Compare the tradeoffs of developing a program in a type-safe language vs other types of programming languages. [Analyzing]

**CYB-17 Implement programs that properly handle exceptions and error conditions.** [Applying] See also SDF-12.

**Emerging Standard**
Explain the importance of writing programs that properly handle exceptions and error conditions. [Understanding]

**Developed Standard**
Implement programs that properly handle exceptions and error conditions. [Applying]

**Highly Developed Standard**
Analyze the implementation of exception handling in programs and error conditions. [Analyzing]

**CYB-18 Examine the need to update software to fix security vulnerabilities.** [Analyzing]

**Emerging Standard**
Explain the need to update software. [Understanding]

**Developed Standard**
Examine the need to update software to fix security vulnerabilities. [Analyzing]

**Highly Developed Standard**
Justify the need to update software to fix security vulnerabilities. [Evaluating]

**CYB-19 Examine likely attack types against standalone and networked systems.** [Analyzing]

**Emerging Standard**
Summarize likely attack types against software systems. [Understanding]

**Developed Standard**
Examine likely attack types against standalone and networked systems. [Analyzing]

**Highly Developed Standard**
Assess likely attack types against standalone and networked systems. [Evaluating]

**CYB-20 Illustrate the key principles of social engineering, including membership and trust.** [Applying]
See also SP-04.

**Emerging Standard**
Discuss some of the key principles of social engineering. [Understanding]

**Developed Standard**
Illustrate the key principles of social engineering, including membership and trust. [Applying]

**Highly Developed Standard**
Outline the key principles of social engineering, including membership and trust. [Analyzing]

**CYB-21 Describe key terms in cryptology, including cryptography, cryptanalysis, cipher, cryptographic**
algorithm, and public key infrastructure. [Understanding]

Emerging Standard
Define some key terms in cryptology. [Remembering]
Developed Standard
Describe key terms in cryptology, including cryptography, cryptanalysis, cipher, cryptographic algorithm, and public key infrastructure. [Understanding]
Highly Developed Standard
Categorize key terms in cryptology, including cryptography, cryptanalysis, cipher, and cryptographic algorithm. [Analyzing]

**CYB-22 Use a variety of ciphers to encrypt plaintext into ciphertext. [Applying]**

Emerging Standard
Describe basic methods for transforming plaintext into ciphertext. [Understanding]
Developed Standard
Use a variety of ciphers to encrypt plaintext into ciphertext. [Applying]
Highly Developed Standard
Design a simple program to encrypt plaintext into ciphertext. [Creating]

**CYB-23 Apply cryptographic hash functions for authentication and data integrity. [Applying]**

Emerging Standard
Summarize the use of cryptographic hash functions for authentication and data integrity. [Understanding]
Developed Standard
Apply cryptographic hash functions for authentication and data integrity. [Applying]
Highly Developed Standard
Deconstruct a cryptographic hash function used for authentication and data integrity. [Analyzing]

**CYB-24 Contrast symmetric and asymmetric encryption in relation to securing electronic communications and transactions. [Analyzing]**

Emerging Standard
Exemplify the difference between symmetric and asymmetric encryption. [Understanding]
Developed Standard
Contrast symmetric and asymmetric encryption in relation to securing electronic communications and transactions. [Analyzing]
Highly Developed Standard
Design a security solution for a given scenario that integrates symmetric encryption with asymmetric encryption. [Creating]

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

Emerging Standard
Identify one or more browser and web security model concepts. [Remembering]
Developed Standard
Explain browser and web security model concepts including same-origin policy, web sessions, and secure communication channels such as TLS. [Understanding]
Highly Developed Standard
Apply browser and web security model concepts including same-origin policy, web sessions, and secure communication channels such as TLS. [Applying]

**DS-01 Explain with examples the basic terminology of functions, relations, and sets. [Understanding]**

Emerging Standard
Identify the defining features of functions, relations, and sets. [Remembering]
Developed Standard
Explain with examples the basic terminology of functions, relations, and sets. [Understanding]
Highly Developed Standard
Use function, relations, and set terminology in a correct and meaningful way. [Applying]

**DS-02 Perform the operations associated with sets, functions, and relations. [Applying]**

Emerging Standard
Describe the operations associated with sets, functions, and relations. [Understanding]
Developed Standard
Perform the operations associated with sets, functions, and relations. [Applying]
Highly Developed Standard
Compare the operations of sets, functions, and relations. [Analyzing]

**DS-03 Compare practical examples to the appropriate set, function, or relation model, and interpret the associated operations and terminology in context. [Analyzing]**

Emerging Standard
Implement a solution to a programming problem using a specific set, function, or relation model. [Applying]
Developed Standard
Compare practical examples to the appropriate set, function, or relation model, and interpret the associated operations and terminology in context. [Analyzing]
Highly Developed Standard
Justify the choice of a specific set, function, or relation model. [Evaluating]

**DS-04 Convert logical statements from informal language to propositional and predicate logic expressions. [Understanding]**

Emerging Standard
Recognize the relationship between logical statements from informal language and propositional and predicate logic expressions. [Remembering]
Developed Standard
Convert logical statements from informal language to propositional and predicate logic expressions.
[Understanding]
Highly Developed Standard
Produce propositional and predicate logic expressions from a given logical statement from an informal language. [Applying]

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]

Emerging Standard
Describe the steps in formal logic proofs and/or informal logical reasoning to solve real problems.
[Understanding]
Developed Standard
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]
Highly Developed Standard
Compare different logic proofs and informal logical reasoning to determine correct methods to solve real problems. [Analyzing]

DS-06 Use the rules of inference to construct proofs in propositional and predicate logic. [Applying]

Emerging Standard
Discuss the rules of inference to construct proofs in propositional and predicate logic. [Understanding]
Developed Standard
Use the rules of inference to construct proofs in propositional and predicate logic. [Applying]
Highly Developed Standard
Analyze the rules of inference to construct proofs in propositional and predicate logic. [Analyzing]

DS-07 Describe how symbolic logic can be used to model real-life situations or computer applications. [Understanding]

Emerging Standard
List ways that symbolic logic can be used to model real-life situations or computer applications. [Remembering]
Developed Standard
Describe how symbolic logic can be used to model real-life situations or computer applications, such as software analysis, database queries, and algorithms. [Understanding]
Highly Developed Standard
Use symbolic logic to model real-life situations or computer applications. [Applying]

DS-08 Apply formal methods of symbolic propositional and predicate logic, such as calculating validity of formulae and computing normal forms. [Applying]

Emerging Standard
Demonstrate formal methods of symbolic propositional and predicate logic. [Understanding]
Developed Standard
Apply formal methods of symbolic propositional and predicate logic, such as calculating validity of formulae
and computing normal forms. [Applying]
Highly Developed Standard
Distinguish between formal methods of propositional and predicate logic to determine the most effective solutions to a given problem. [Analyzing]

**DS-09 Describe the strengths and limitations of propositional and predicate logic. [Understanding]**

Emerging Standard
List the strengths and limitations of propositional and predicate logic. [Remembering]
Developed Standard
Describe the strengths and limitations of propositional and predicate logic. [Understanding]
Highly Developed Standard
Illustrate the strengths and limitations of propositional and predicate logic. [Applying]

**DS-10 Outline the basic structure of each proof technique, including direct proof, proof by contradiction, and induction. [Analyzing]**

Emerging Standard
Use the basic structure of each proof technique to solve a problem. [Applying]
Developed Standard
Outline the basic structure of each proof technique, including direct proof, proof by contradiction, and induction. [Analyzing]
Highly Developed Standard
Choose the most effective proof technique to solve a problem. [Evaluating]

**DS-11 Apply each of the proof techniques (direct proof, proof by contradiction, and induction) correctly in the construction of a sound argument. [Applying]**

Emerging Standard
Demonstrate each of the proof techniques by correctly constructing a sound argument. [Understanding]
Developed Standard
Apply each of the proof techniques (direct proof, proof by contradiction, and induction) correctly in the construction of a sound argument. [Applying]
Highly Developed Standard
Use each of the proof techniques correctly in the construction of a sound argument. [Applying]

**DS-12 Deduce the best type of proof for a given problem. [Analyzing]**

Emerging Standard
Compare the different proof methods. [Analyzing]
Developed Standard
Deduce the best type of proof for a given problem. [Analyzing]
Highly Developed Standard
Construct a correct proof using the best method for a given problem. [Creating]
DS-13 Explain the parallels between ideas of mathematical and/or structural induction to recursion and recursively defined structures. [Understanding]

Emerging Standard
Identify the parallels between ideas of mathematical and/or structural induction to recursion and recursively defined structures. [Remembering]

Developed Standard
Explain the parallels between ideas of mathematical and/or structural induction to recursion and recursively defined structures. [Understanding]

Highly Developed Standard
Illustrate the parallels between ideas of mathematical and/or structural induction to recursion and recursively defined structures. [Applying]

DS-14 Explain the relationship between weak and strong induction and give examples of the appropriate use of each. [Understanding]

Emerging Standard
Identify the relationship between weak and strong induction. [Remembering]

Developed Standard
Explain the relationship between weak and strong induction and give examples of the appropriate use of each. [Understanding]

Highly Developed Standard
Solve problems using both weak and strong induction. [Applying]

DS-15 Apply counting arguments, including sum and product rules, inclusion-exclusion principle and arithmetic/geometric progressions. [Applying]

Emerging Standard
Describe counting arguments. [Understanding]

Developed Standard
Apply counting arguments, including sum and product rules, inclusion-exclusion principle and arithmetic/geometric progressions. [Applying]

Highly Developed Standard
Outline counting arguments. [Analyzing]

DS-16 Apply the pigeonhole principle in the context of a formal proof. [Applying]

Emerging Standard
Demonstrate the pigeonhole principle. [Understanding]

Developed Standard
Apply the pigeonhole principle in the context of a formal proof. [Applying]

Highly Developed Standard
Analyze the pigeonhole principle in the context of a formal proof. [Analyzing]
DS-17 Calculate permutations and combinations of a set, and interpret the meaning in the context of the particular application. [Applying]

Emerging Standard
Explain the calculation of permutations and combinations of a set. [Understanding]

Developed Standard
Calculate permutations and combinations of a set, and interpret the meaning in the context of the particular application. [Applying]

Highly Developed Standard
 Discriminate between computation of st permutations and combinations. [Analyzing]

DS-18 Compare real-world applications appropriate to counting formalisms. [Analyzing]

Emerging Standard
Use counting formalisms to solve real-world applications. [Applying]

Developed Standard
Compare real-world applications appropriate to counting formalisms, such as determining the number of ways to arrange people around a table, subject to constraints on the seating arrangement. [Analyzing]

Highly Developed Standard
Choose appropriate counting formalisms to solve real-world applications. [Evaluating]

DS-19 Solve a variety of basic recurrence relations. [Applying]

Emerging Standard
Demonstrate a variety of basic recurrence relations. [Understanding]

Developed Standard
Solve a variety of basic recurrence relations. [Applying]

Highly Developed Standard
Compare a variety of basic recurrence relations. [Analyzing]

DS-20 Analyze a problem to determine underlying recurrence relations. [Analyzing]

Emerging Standard
Carry out a problem with an underlying recurrence relation. [Applying]

Developed Standard
Analyze a problem to determine underlying recurrence relations. [Analyzing]

Highly Developed Standard
Evaluate a problem with an underlying recurrence relation. [Evaluating]

DS-21 Perform computations involving modular arithmetic. [Applying]

Emerging Standard
Discuss computations involving modular arithmetic. [Understanding]

Developed Standard
Perform computations involving modular arithmetic. [Applying]
Highly Developed Standard
Examine computations involving modular arithmetic. [Analyzing]

**DS-22 Illustrate the basic terminology of graph theory including properties and special cases for each type of graph/tree. [Applying]**

Emerging Standard
Describe the basic terminology of graph theory. [Understanding]
Developed Standard
Illustrate the basic terminology of graph theory including properties and special cases for each type of graph/tree. [Applying]
Highly Developed Standard
Outline the basic terminology of graph theory. [Analyzing]

**DS-23 Demonstrate different traversal methods for trees and graphs, including pre-, post-, and in-order traversal of trees. [Understanding]**

Emerging Standard
List the different traversal methods for trees and graphs [Remembering]
Developed Standard
Demonstrate different traversal methods for trees and graphs, including pre-, post-, and in-order traversal of trees. [Understanding]
Highly Developed Standard
Execute different traversal methods for trees and graphs. [Applying]

**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]**

Emerging Standard
Discuss a variety of real-world problems in computer science using appropriate forms of graphs and trees. [Understanding]
Developed Standard
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]
Highly Developed Standard
Distinguish between real-world problems solvable by using graphs and trees. [Analyzing]

**DS-25 Implement and use balanced trees and B-trees. [Applying]**

Emerging Standard
Explain balanced trees and B-trees. [Understanding]
Developed Standard
Implement and use balanced trees and B-trees. [Applying]
Highly Developed Standard
Analyze the use of balanced trees and B-trees. [Analyzing]

**DS-26 Implement graph algorithms. [Applying]**

Emerging Standard
Classify graph algorithms. [Understanding]
Developed Standard
Implement graph algorithms, such as graph search, union-find, minimum spanning trees, and shortest paths. [Applying]
Highly Developed Standard
Categorize different implementations of graph algorithms. [Analyzing]

**DS-27 Demonstrate how concepts from graphs and trees appear in data structures, algorithms, proof techniques (structural induction), and counting. [Understanding]**

Emerging Standard
Identify how concepts from graphs and trees appear in data structures, algorithms, proof techniques, and counting. [Remembering]
Developed Standard
Demonstrate how concepts from graphs and trees appear in data structures, algorithms, proof techniques (structural induction), and counting. [Understanding]
Highly Developed Standard
Implement data structures, algorithms, proof techniques, and counting using graphs and trees. [Applying]

**DS-28 Describe binary search trees and AVL trees. [Understanding]**

Emerging Standard
Define binary search and AVL trees. [Remembering]
Developed Standard
Describe binary search trees and AVL trees. [Understanding]
Highly Developed Standard
Apply binary search and AVL trees. [Applying]

**DS-29 Explain complexity in the ideal and in the worst-case scenario for both implementations. [Understanding]**

Emerging Standard
State complexity in the ideal and in the worst-case scenario for both implementations. [Remembering]
Developed Standard
Explain complexity in the ideal and in the worst-case scenario for both implementations. [Understanding]
Highly Developed Standard
Calculate complexity in the ideal and in the worst-case scenario for both implementations. [Applying]
DS-30 Calculate probabilities of events and expectations of random variables for elementary problems. [Applying]

Emerging Standard
Exemplify probabilities of events and expectations of random variables for elementary problems such as games of chance. [Understanding]

Developed Standard
Calculate probabilities of events and expectations of random variables for elementary problems such as games of chance. [Applying]

Highly Developed Standard
Examine probabilities of events and expectations of random variables for elementary problems such as games of chance. [Analyzing]

DS-31 Differentiate between dependent and independent events. [Understanding]

Emerging Standard
Identify dependent and independent events. [Remembering]

Developed Standard
Differentiate between dependent and independent events. [Understanding]

Highly Developed Standard
Illustrate dependent and independent events. [Applying]

DS-32 Explain the significance of binomial distribution in probabilities. [Understanding]

Emerging Standard
Recognize the notation and parameters that a binomial distribution has. [Remembering]

Developed Standard
Explain the significance of binomial distribution in probabilities. [Understanding]

Highly Developed Standard
Calculate the probabilities from a binomial distribution. [Applying]

DS-33 Apply Bayes Theorem to determine conditional probabilities in a problem. [Applying]

Emerging Standard
Explain Bayes Theorem to determine conditional probabilities in a problem. [Understanding]

Developed Standard
Apply Bayes Theorem to determine conditional probabilities in a problem. [Applying]

Highly Developed Standard
Outline Bayes Theorem to determine conditional probabilities in a problem. [Analyzing]

DS-34 Apply the tools of probability to solve problems. [Applying]

Emerging Standard
Discuss the tools of probability to solve problems such as the average case analysis of algorithms or analyzing
hashing. [Understanding]

Developed Standard
Apply the tools of probability to solve problems such as the average case analysis of algorithms or analyzing hashing. [Applying]

Highly Developed Standard
Analyze the tools of probability in solving problems such as the average case analysis of algorithms or analyzing hashing. [Analyzing]

**GV-01 Compare transformation and changes in dimension and coordinate systems for 2D and 3D design.** [Analyzing]

Emerging Standard
Explain how to use dimensions and coordinate systems. [Understanding]

Developed Standard
Compare transformation and changes in dimension and coordinate systems for 2D and 3D design. [Analyzing]

Highly Developed Standard
Evaluate the uses of both 3D coordinate systems and 2D planar systems with respect to computer graphics. [Evaluating]

**GV-02 Demonstrate common uses of digital presentation to human senses.** [Understanding]

Emerging Standard
Identify common uses of digital presentation to human senses. [Remembering]

Developed Standard
Demonstrate common uses of digital presentation to human senses, such as computer graphics, sound, and haptic devices. [Understanding]

Highly Developed Standard
Categorize types of digital presentation to human senses. [Analyzing]

**GV-03 Illustrate color models and their use in computer graphics.** [Applying]

Emerging Standard
Describe color models and their use in computer graphics. [Understanding]

Developed Standard
Illustrate color models and their use in computer graphics. [Applying]

Highly Developed Standard
Contrast color models and their use in computer graphics. [Analyzing]

**GV-04 Analyze image types according to output choices.** [Analyzing]

Emerging Standard
Differentiate multimedia file types, resolution needs, conversion, and appropriate use. [Understanding]

Developed Standard
Analyze image types according to output choices. [Analyzing]

Highly Developed Standard
Evaluate multiple multimedia files based on given criteria. [Evaluating]

**GV-05 Perform information hiding through steganography in images, messages, videos, or other media files. [Applying]**

Emerging Standard  
Demonstrate information hiding through steganography. [Understanding]  
Developed Standard  
Perform information hiding through steganography in images, messages, videos, or other media files. [Applying]  
Highly Developed Standard  
Choose appropriate steganography technique to conceal information. [Evaluating]

**HCI-01 Analyze the importance of human-centered software. [Analyzing]**

Emerging Standard  
Discuss the importance of human-centered software development. [Understanding]  
Developed Standard  
Analyze the importance of human-centered software. [Analyzing]  
Highly Developed Standard  
Defend the importance of human-centered software. [Evaluating]

**HCI-02 Implement a simple usability test for an existing software application. [Applying]**

Emerging Standard  
Summarize usability testing. [Understanding]  
Developed Standard  
Implement a simple usability test for an existing software application. [Applying]  
Highly Developed Standard  
Design a usability test for an existing software application [Creating]

**HCI-03 Examine the issues of trust in HCI, including examples of both high and low trust systems. [Analyzing]**

Emerging Standard  
Demonstrate design elements that make a human-computer interface trustworthy. [Understanding]  
Developed Standard  
Examine the issues of trust in HCI, including examples of both high and low trust systems. [Analyzing]  
Highly Developed Standard  
Critique interface designs between high trust and low trust. [Evaluating]

**HCI-04 Write a simple application that uses a modern graphical user interface. [Applying]**

Emerging Standard
Summarize the components of a modern graphical user interface. [Understanding]
Developed Standard
Write a simple application that uses a modern graphical user interface. [Applying]
Highly Developed Standard
Examine a complete application that uses a modern graphical interface and documentation. [Analyzing]

**HCI-05 Use at least one national or international user interface design standard in a simple application.** [Applying] See also SP-02.

Emerging Standard
Identify national and international user interface design standards. [Remembering]
Developed Standard
Use at least one national or international user interface design standard in a simple application, such as U.S. ADA Standards. [Applying]
Highly Developed Standard
Compare among national and international user interface design standards. [Analyzing]

**HCI-06 Analyze the interaction between a security mechanism and its usability.** [Analyzing]

Emerging Standard
Discuss potential usability issues related to a security mechanism. [Understanding]
Developed Standard
Analyze the interaction between a security mechanism and its usability. [Analyzing]
Highly Developed Standard
Design a user interface that balances the tradeoffs between usability and security. [Creating]

**IM-01 Contrast the difference between information and data.** [Analyzing]

Emerging Standard
Explain information and data. [Understanding]
Developed Standard
Contrast the difference between information and data. [Analyzing]
Highly Developed Standard
Critique the attributes of both information and data. [Evaluating]

**IM-02 Describe the advantages and disadvantages of central organizational control over data.** [Understanding]

Emerging Standard
List some advantages and disadvantages of central organizational control over data. [Remembering]
Developed Standard
Describe the advantages and disadvantages of central organizational control over data. [Understanding]
Highly Developed Standard
Examine the advantages and disadvantages of central organizational control over data. [Analyzing]
IM-03 Investigate contingency planning with respect to business continuity, disaster recovery and incident response. [Applying]

Emerging Standard
Describe contingency planning with respect to business continuity, disaster recovery and incident response. [Understanding]

Developed Standard
Investigate contingency planning with respect to business continuity, disaster recovery and incident response. [Applying]

Highly Developed Standard
Compare contingency plans of various size organizations with respect to business continuity, disaster recovery and incident response. [Analyzing]

IM-04 Describe proven techniques to secure data and information. [Understanding]

Emerging Standard
List proven techniques used to secure data and information. [Remembering]

Developed Standard
Describe proven techniques to secure data and information. [Understanding]

Highly Developed Standard
Implement specific proven techniques to secure data and information. [Applying]

IM-05 Describe approaches to scale up information systems. [Understanding]

Emerging Standard
List approaches to scale up information systems. [Remembering]

Developed Standard
Describe approaches to scale up information systems. [Understanding]

Highly Developed Standard
Compare several approaches to scale up information systems. [Analyzing]

IM-06 Explain the characteristics that distinguish the database approach from the approach of programming with data files. [Understanding]

Emerging Standard
Identify the characteristics that distinguish the database approach from the approach of programming with data files. [Remembering]

Developed Standard
Explain the characteristics that distinguish the database approach from the approach of programming with data files. [Understanding]

Highly Developed Standard
Contrast the characteristics that distinguish the database approach from the approach of programming with data files. [Analyzing]
IM-07 Diagram the components of a database system and give examples of their use. [Applying]

Emerging Standard
Identify the components of a database system. [Remembering]

Developed Standard
Diagram the components of a database system and give examples of their use. [Applying]

Highly Developed Standard
Organize the components of a database system into a secure, functioning system. [Analyzing]

IM-08 Explain the concept of data independence and its importance in a database system. [Understanding]

Emerging Standard
Define the concept of data independence and its importance in a database system. [Remembering]

Developed Standard
Explain the concept of data independence and its importance in a database system. [Understanding]

Highly Developed Standard
Examine the concept of data independence and its importance in a database system. [Analyzing]

IM-09 Formulate queries in SQL or a similar query language to elicit information from a database. [Creating]

Emerging Standard
Edit queries in SQL or a similar query language to elicit information from a database. [Applying]

Developed Standard
Formulate queries in SQL or a similar query language to elicit information from a database. [Creating]

Highly Developed Standard
Formulate complex 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]

Emerging Standard
Identify vulnerabilities and failure scenarios in information systems. [Remembering]

Developed Standard
Investigate vulnerabilities and failure scenarios in database systems, such as SQL injection and cross-site scripting. [Applying]

Highly Developed Standard
Examine vulnerabilities and failure scenarios in information systems. [Analyzing]

IM-11 Contrast appropriate data models, including internal structures for different data types. [Analyzing]
Emerging Standard
Diagram appropriate data models, including internal structures for different data types. [Applying]

Developed Standard
Contrast appropriate data models, including internal structures for different data types. [Analyzing]

Highly Developed Standard
Evaluate appropriate data models, including internal structures for different data types. [Evaluating]

IM-12 Diagram a relational data model for a given scenario that addresses security and privacy concerns. [Applying]

Emerging Standard
Describe a relational data model for a given scenario that addresses security and privacy concerns. [Understanding]

Developed Standard
Diagram a relational data model for a given scenario that addresses security and privacy concerns. [Applying]

Highly Developed Standard
Analyze a relational data model for a given scenario that addresses security and privacy concerns. [Analyzing]

IM-13 Describe the differences among relational data models and other models. [Understanding]

Emerging Standard
Recognize the differences among relational data models and other models. [Remembering]

Developed Standard
Describe the differences among relational data models and other models such as Object-Oriented, JSON, NoSQL. [Understanding]

Highly Developed Standard
Analyze the differences among relational data models and other models. [Analyzing]

NC-01 Diagram the basic structure of the Internet. [Applying]

Emerging Standard
Explain the basic structure of the Internet. [Understanding]

Developed Standard
Diagram the basic structure of the Internet. [Applying]

Highly Developed Standard
Analyze the basic structure of the internet. [Analyzing]

NC-02 Describe the layered structure of a typical networked architecture, including routing and switching. [Understanding]

Emerging Standard
Label the components of a typical networked architecture. [Remembering]
Developed Standard
Describe the layered structure of a typical networked architecture, including routing and switching. [Understanding]

Highly Developed Standard
Diagram the layered structure of a typical networked architecture. [Applying]

NC-03 Diagram the layers of the OSI model, including associated protocols. [Applying]

Emerging Standard
Describe the layers of the OSI model. [Understanding]

Developed Standard
Diagram the layers of the OSI model, including associated protocols. [Applying]

Highly Developed Standard
Compare the layers of the OSI model with the TCP/IP model. [Analyzing]

NC-04 Categorize the principles used for naming schemes and resource location. [Analyzing]

Emerging Standard
Summarize principles used for naming schemes and resource location. [Understanding]

Developed Standard
Categorize the principles used for naming schemes and resource location. [Analyzing]

Highly Developed Standard
Evaluate the principles used for naming schemes and resource location. [Evaluating]

NC-05 Implement a simple distributed network application. [Applying]

Emerging Standard
Identify the components of a simple distributed network application. [Remembering]

Developed Standard
Implement a simple distributed network application. [Applying]

Highly Developed Standard
Integrate a simple distributed network application within a server program to exchange data with a client. [Analyzing]

NC-06 Describe security concerns in designing applications for use over wireless networks. [Understanding]

Emerging Standard
Recognize security concerns in designing applications for use over wireless networks. [Remembering]

Developed Standard
Describe security concerns in designing applications for use over wireless networks. [Understanding]

Highly Developed Standard
Illustrate security concerns in designing applications for use over wireless networks. [Applying]
NC-07 Illustrate secure connectivity among networked applications. [Applying]

Emerging Standard
Explain secure connectivity among networked applications. [Understanding]
Developed Standard
Illustrate secure connectivity among networked applications, such as SSH, HTTPS, SFTP. [Applying]
Highly Developed Standard
Critique secure connectivity among networked applications. [Evaluating]

NC-08 Explain the advantages and disadvantages of using virtualized infrastructure in cloud computing. [Understanding] See also PD-05

Emerging Standard
List the advantages and disadvantages of using virtualized infrastructure in cloud computing. [Remembering]
Developed Standard
Explain the advantages and disadvantages of using virtualized infrastructure in cloud computing. [Understanding]
Highly Developed Standard
Investigate the advantages and disadvantages of using virtualized infrastructure in cloud computing. [Applying]

OS-01 Examine major objectives, functions, features, and concepts of modern operating systems. [Analyzing]

Emerging Standard
Describe objectives, functions, features, and concepts of modern operating systems. [Understanding]
Developed Standard
Examine major objectives, functions, features, and concepts of modern operating systems. [Analyzing]
Highly Developed Standard
Assess major objectives, functions, and concepts of modern operating systems. [Evaluating]

OS-02 Compare prevailing types of operating systems. [Analyzing]

Emerging Standard
Investigate prevailing types of operating systems. [Applying]
Developed Standard
Compare prevailing types of operating systems, such as networked, mobile, embedded, and real-time. [Analyzing]
Highly Developed Standard
Assess prevailing types of operating systems. [Evaluating]

OS-03 Illustrate potential threats to operating systems and appropriate security measures. [Applying]
Emerging Standard
Describe potential threats to operating systems and appropriate security measures. [Understanding]

Developed Standard
Illustrate potential threats to operating systems and appropriate security countermeasures. [Applying]

Highly Developed Standard
Examine potential threats to operating systems and appropriate security countermeasures. [Analyzing]

**OS-04 Diagram the interaction of an Application Programming Interface (API) with an operating system. [Applying]**

Emerging Standard
Summarize the interaction of an Application Programming Interface (API) with an operating system. [Understanding]

Developed Standard
Diagram the interaction of an Application Programming Interface (API) with an operating system. [Applying]

Highly Developed Standard
Test the interaction of an Application Programming Interface (API) with an operating system. [Evaluating]

**OS-05 Illustrate how computing resources are used by applications and managed by the operating system. [Applying]**

Emerging Standard
Exemplify how computing resources are used by applications and managed by the operating system. [Understanding]

Developed Standard
Illustrate how computing resources are used by applications and managed by the operating system. [Applying]

Highly Developed Standard
Test how computing resources are used by applications and managed by the operating system. [Evaluating]

**OS-06 Manipulate a device list or driver I/O queue. [Applying]**

Emerging Standard
Explain the purpose of a device list and driver I/O queue. [Understanding]

Developed Standard
Manipulate a device list or driver I/O queue. [Applying]

Highly Developed Standard
Categorize device types of a modern operating system. [Analyzing]

**OS-07 Investigate the need for concurrency within an operating system. [Applying] See also PD-01.**

Emerging Standard
Describe the need for concurrency within an operating system. [Understanding]

Developed Standard
Investigate the need for concurrency within an operating system. [Applying]

Highly Developed Standard
Analyze the need for concurrency within an operating system. [Analyzing]

OS-08 Illustrate the principles of memory management. [Applying]
Emerging Standard
Describe the principles of memory management. [Understanding]
Developed Standard
Illustrate the principles of memory management, such as memory hierarchy and allocation, tradeoffs, and caching. [Applying]
Highly Developed Standard
Analyze the principles of memory management. [Analyzing]

OS-09 Illustrate the concepts of virtual memory, including paging, thrashing, and partitioning. [Applying]
Emerging Standard
Describe the concepts of virtual memory, including paging, thrashing, and partitioning [Understanding]
Developed Standard
Illustrate the concepts of virtual memory, including paging, thrashing, and partitioning. [Applying]
Highly Developed Standard
Examine the concepts of virtual memory, including paging, thrashing, and partitioning [Analyzing]

OS-10 Investigate the features and limitations of an operating system used to provide protection and security. [Applying]
Emerging Standard
Explain the features and limitation of an operating system used to provide protection and security. [Understanding]
Developed Standard
Investigate the features and limitation of an operating system used to provide protection and security. [Applying]
Highly Developed Standard
Examine the features and limitation of an operating system used to provide protection and security. [Analyzing]

OS-11 Use mechanisms available in an operating system to control access to resources. [Applying]
Emerging Standard
Summarize the mechanisms available in an operating system to control access to resources. [Understanding]
Developed Standard
Use the mechanisms available in an operating system to control access to resources. [Applying]
Highly Developed Standard
Test the mechanisms available in an operating system to control access to resources. [Evaluating]
OS-12 Analyze the concept of virtualization with respect to hardware and software. [Analyzing]

Emerging Standard
Investigate the concept of virtualization with respect to hardware and software. [Applying]
Developed Standard
Analyze the concept of virtualization with respect to hardware and software. [Analyzing]
Highly Developed Standard
Assess a given implementation of virtualization with respect to hardware and software. [Evaluating]

OS-13 Diagram the physical hardware devices and the virtual devices maintained by an operating system. [Applying]

Emerging Standard
Explain the relationship between the physical hardware devices and virtual devices maintained by an operating system. [Understanding]
Developed Standard
Diagram the physical hardware devices and the virtual devices maintained by an operating system. [Applying]
Highly Developed Standard
Distinguish between the physical hardware devices and virtual devices used by an operating system for a given implementation of virtualization. [Analyzing]

PD-01 Analyze the goals of parallelism and concurrency. [Analyzing] See also OS-07 and SF-08.

Emerging Standard
Differentiate the goal of parallelism, such as throughput, from the goal of concurrency, such as controlling access to shared resources. [Understanding]
Developed Standard
Analyze the goals of parallelism and concurrency. [Analyzing]
Highly Developed Standard
Evaluate the performance of a given program that was implemented using parallelism and concurrency techniques. [Evaluating]

PD-02 Implement various programming constructs for synchronization. [Applying]

Emerging Standard
Summarize various programming constructs for synchronization. [Understanding]
Developed Standard
Implement various programming constructs for synchronization. [Applying]
Highly Developed Standard
Integrate synchronization routines/techniques into a non-synchronized programming constructs. [Analyzing]

PD-03 Contrast low-level data races with higher level races. [Analyzing]

Emerging Standard
Differentiate low-level data races from higher level races. [Understanding]
Developed Standard
Contrast low-level data races with higher level races. [Analyzing]
Highly Developed Standard
Create a low-level data race among two concurrent threads. [Creating]

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

Emerging Standard
Explain mutual exclusion in order to avoid race conditions. [Understanding]
Developed Standard
Implement mutual exclusion in order to avoid race conditions that could cause security vulnerabilities. [Applying]
Highly Developed Standard
Categorize critical and noncritical race conditions. [Analyzing]

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

Emerging Standard
Describe the challenges and concerns related to security and privacy in Cloud computing. [Understanding]
Developed Standard
Investigate the challenges and concerns related to security and privacy in Cloud computing. [Applying]
Highly Developed Standard
Examine the challenges and concerns related to security and privacy in Cloud computing. [Analyzing]

**PL-01 Design a simple class hierarchy using superclasses, subclasses, and abstract classes. [Creating]**

Emerging Standard
Implement a simple class hierarchy using superclasses, subclasses, and abstract classes. [Applying]
Developed Standard
Design a simple class hierarchy using superclasses, subclasses, and abstract classes. [Creating]
Highly Developed Standard
Develop a complex class hierarchy using superclasses, subclasses, and abstract classes. [Creating]

**PL-02 Diagram control flow in a program using dynamic dispatch. [Applying]**

Emerging Standard
Demonstrate control flow that uses dynamic dispatch. [Understanding]
Developed Standard
Diagram control flow in a program using dynamic dispatch. [Applying]
Highly Developed Standard
Contrast control flow that uses a static environment vs a dynamic environment. [Analyzing]
PL-03 Use access and visibility modifiers to secure class data and methods. [Applying]

Emerging Standard
Describe access modifiers to secure class data such as private and protected. [Understanding]
Developed Standard
Use access and visibility modifiers to secure class data and methods, such as private and protected. [Applying]
Highly Developed Standard
Analyze the security effect of using access and visibility modifiers in code. [Analyzing]

PL-04 Implement in code OOP constructs, including encapsulation, abstraction, inheritance, and polymorphism. [Applying]

Emerging Standard
Demonstrate the tenets of OOP, including encapsulation, abstraction, inheritance, and polymorphism. [Understanding]
Developed Standard
Implement in code OOP constructs, including encapsulation, abstraction, inheritance, and polymorphism. [Applying]
Highly Developed Standard
Create a program that utilizes OOP constructs, including encapsulation, abstraction, inheritance, and polymorphism. [Analyzing]

PL-05 Implement algorithms which utilize immutable and mutable variables. [Applying]

Emerging Standard
Discuss how functional languages handle both immutable and mutable variables. [Understanding]
Developed Standard
Implement algorithms which utilize immutable and mutable variables. [Applying]
Highly Developed Standard
Evaluate the efficiency of different algorithms which utilize immutable vs mutable variables. [Evaluating]

PL-06 Contrast functional and object-oriented programming paradigms. [Analyzing]

Emerging Standard
Explain major differences between functional and object-oriented programming paradigms. [Understanding]
Developed Standard
Contrast functional and object-oriented programming paradigms. [Analyzing]
Highly Developed Standard
Appraise functional and object-oriented programming paradigms. [Evaluating]

PL-07 Create an interactive program using an event-driven style. [Creating]

Emerging Standard
Describe advantages of having an event-driven programming style vs a pre-defined programming style.
PL-08 Describe potential security vulnerabilities in event-driven GUI applications. [Understanding]

Emerging Standard
Identify potential security vulnerabilities in event-driven GUI applications. [Remembering]

Developed Standard
Describe potential security vulnerabilities in event-driven GUI applications, such as injection-based attacks. [Understanding]

Highly Developed Standard
Illustrate potential security vulnerabilities in event-driven GUI applications. [Applying]

PL-09 Investigate potential errors detected from both strong-type and weak-type languages. [Applying]

Emerging Standard
Summarize possible errors detected from both strong-type and weak-type languages. [Understanding]

Developed Standard
Investigate possible errors detected from both strong-type and weak-type languages. [Applying]

Highly Developed Standard
Discriminate among errors detected from strong-type and from weak-type languages. [Analyzing]

PL-10 Explain the security implications of a type-safe language for software development. [Understanding]

Emerging Standard
Recognize the security implications of a type-safe language for software development. [Remembering]

Developed Standard
Explain the security implications of a type-safe language for software development. [Understanding]

Highly Developed Standard
Examine the security advantages and disadvantages of a type-safe language for software development. [Analyzing]

SDF-01 Design an algorithm in a programming language to solve a simple problem. [Creating]

Emerging Standard
Implement an algorithm in a programming language to solve a simple problem. [Applying]

Developed Standard
Design an algorithm in a programming language to solve a simple problem. [Creating]

Highly Developed Standard
Design an algorithm in a programming language to solve a complex problem. [Creating]
**SDF-02 Use the techniques of decomposition to modularize a program. [Applying]**

Emerging Standard
Explain program decomposition. [Understanding]

Developed Standard
Use the techniques of decomposition to modularize a program. [Applying]

Highly Developed Standard
Analyze code to see how decomposition techniques were used. [Analyzing]

**SDF-03 Compare multiple algorithms for a given problem. [Analyzing]**

Emerging Standard
Investigate multiple algorithms for a given problem. [Applying]

Developed Standard
Compare multiple algorithms for a given problem. [Analyzing]

Highly Developed Standard
Evaluate the strengths and weaknesses of multiple algorithms for a problem. [Evaluating]

**SDF-04 Create simple programs that use abstract data types (ADTs). [Creating]**

Emerging Standard
Implement simple programs that use abstract data types (ADTs). [Applying]

Developed Standard
Create simple programs that use abstract data types (ADTs). [Creating]

Highly Developed Standard
Write complex 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.**

Emerging Standard
Summarize potential vulnerabilities in programming code. [Understanding]

Developed Standard
Investigate potential vulnerabilities in provided programming code. [Applying]

Highly Developed Standard
Choose a solution to mitigate vulnerabilities in programming code. [Evaluating]

**SDF-06 Create programs which use defensive programming techniques, including input validation, type checking, and protection against buffer overflow. [Creating] See also CYB-15.**

Emerging Standard
Investigate defensive programming techniques. [Applying]

Developed Standard
Create programs which use defensive programming techniques, including input validation, type checking, and
Create complex programs which use defensive programming techniques, including input validation, type checking, and protection against buffer overflow. [Creating]

SDF-07 Create code in a programming language that includes primitive data types, references, variables, expressions, assignments, I/O, control structures, and functions. [Creating]

Emerging Standard
Implement code in a programming language that includes primitive data types, references, variables, expressions, assignments, I/O, control structures, and functions. [Applying]

Developed Standard
Create code in a programming language that includes primitive data types, references, variables, expressions, assignments, I/O, control structures, and functions. [Creating]

Highly Developed Standard
Create complex programs that include 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]

Emerging Standard
Implement a simple program that uses persistence to save data across multiple executions. [Applying]

Developed Standard
Create a simple program that uses persistence to save data across multiple executions. [Creating]

Highly Developed Standard
Create a complex program that uses persistence to save data across multiple executions. [Creating]

SDF-09 Create a simple program that uses recursion. [Creating]

Emerging Standard
Implement a simple program that uses recursion. [Applying]

Developed Standard
Create a simple program that uses recursion. [Creating]

Highly Developed Standard
Develop a complex program that includes various types of recursive techniques, such as binary, tail, and natural recursion. [Creating]

SDF-10 Create simple programs that include each of the following data structures: lists, stacks, queues, hash tables, graphs and trees. [Creating]

Emerging Standard
Implement programs that include each of the following data structures: lists, stacks, queues, hash tables, graphs and trees. [Applying]

Developed Standard
Create simple programs that include each of the following data structures: lists, stacks, queues, hash tables,
Create complex 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]

**Emerging Standard**
Investigate the efficiency of basic operations across various data structures. [Applying]

**Developed Standard**
Compare the efficiency of basic operations, such as insertion and deletion, across various data structures. [Analyzing]

**Highly Developed Standard**
Critique the efficiency of basic operations across various data structures. [Evaluating]

**SDF-12 Investigate common coding errors that introduce security vulnerabilities.** [Applying]

**Emerging Standard**
Describe common coding errors that expose security vulnerabilities. [Understanding]

**Developed Standard**
Investigate common coding errors that introduce security vulnerabilities, such as buffer overflows, integer errors, and memory leaks. [Applying]

**Highly Developed Standard**
Test for common coding errors that introduce security vulnerabilities and the associated techniques for securing the code. [Evaluating]

**SDF-13 Implement refactoring within given program components.** [Applying]

**Emerging Standard**
Recognize refactoring opportunities within given program components. [Remembering]

**Developed Standard**
Implement refactoring within given program components. [Applying]

**Highly Developed Standard**
Create program components utilizing refactoring. [Applying]

**SDF-14 Analyze programming code that utilizes preconditions, postconditions, and invariants.** [Analyzing]

**Emerging Standard**
Describe programming by contract. [Understanding]

**Developed Standard**
Analyze programming code that utilizes preconditions, postconditions, and invariants. [Analyzing]

**Highly Developed Standard**
Create a program utilizing preconditions, postconditions, and invariants. [Creating]
SDF-15 Apply a variety of strategies to test and debug programs. [Applying]

Emerging Standard
Explain strategies to test and debug programs. [Understanding]

Developed Standard
Apply a variety of strategies to test and debug programs, such as unit testing and test-case generation. [Applying]

Highly Developed Standard
Analyze a variety of strategies to test and debug programs. [Analyzing]

SDF-16 Use an integrated development environment (IDE) to create, execute, test, and debug secure programs. [Applying]

Emerging Standard
Discuss the benefits of using an integrated development environment (IDE) to create, execute, test, and debug secure programs. [Understanding]

Developed Standard
Use an integrated development environment (IDE) to create, execute, test, and debug secure programs. [Applying]

Highly Developed Standard
Compare integrated development environments (IDEs) for a given programming language. [Analyzing]

SDF-17 Use standard libraries for a given programming language. [Applying]

Emerging Standard
Describe standard libraries for a given programming language. [Understanding]

Developed Standard
Use standard libraries for a given programming language. [Applying]

Highly Developed Standard
Choose appropriate components from standard libraries to solve a given problem. [Evaluating]

SDF-18 Apply consistent documentation and program style standards. [Applying]

Emerging Standard
Explain the reasons for using consistent documentation and program style standards. [Understanding]

Developed Standard
Apply consistent documentation and program style standards. [Applying]

Highly Developed Standard
Assess documentation and program style in a given program. [Evaluating]

SDF-19 Carry out a code review on a program component using a provided security checklist. [Applying]
Emerging Standard
Explain the process of a code review. [Understanding]

Developed Standard
Carry out a code review on a program component using a provided security checklist. [Applying]

Highly Developed Standard
Organize a team code review on a program component using a provided security checklist. [Analyzing]

SE-01 Diagram how software interacts with various systems, including information management, embedded, process control, and communications systems. [Applying]

Emerging Standard
Describe how software interacts with various systems including information management, embedded, process control, and communications systems. [Understanding]

Developed Standard
Diagram how software interacts with various systems, including information management, embedded, process control, and communications systems. [Applying]

Highly Developed Standard
Assess how software interacts with various systems, including information management, embedded, process control, and communications systems. [Evaluating]

SE-02 Compare the features of various process models. [Analyzing]

Emerging Standard
Describe the features of a process model. [Understanding]

Developed Standard
Compare the features of various process models, such as waterfall, iterative, and agile. [Analyzing]

Highly Developed Standard
Critique various process models. [Evaluating]

SE-03 Diagram the phases of the secure software development lifecycle (SecSDLC). [Applying]

Emerging Standard
Exemplify the phases the software development lifecycle (SecSDLC). [Understanding]

Developed Standard
Diagram the phases of the secure software development lifecycle (SecSDLC). [Applying]

Highly Developed Standard
Examine the phases of the secure software development lifecycle (SecSDLC). [Analyzing]

SE-04 Illustrate common behaviors that contribute to the effective functioning of a team. [Applying]

Emerging Standard
Describe common behaviors that contribute to the effective functioning of a team. [Understanding]
Developed Standard
Illustrate common behaviors that contribute to the effective functioning of a team, such as good communication skills. [Applying]

Highly Developed Standard
Examine common behaviors that contribute to the effective functioning of a team. [Analyzing]

SE-05 Investigate the risks in using third-party applications, software tools, and libraries. [Applying]

Emerging Standard
Explain the risks in using third-party code. [Understanding]

Developed Standard
Investigate the risks in using third-party applications, software tools, and libraries. [Applying]

Highly Developed Standard
Evaluate the risks in using third-party applications, software tools, and libraries. [Evaluating]

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

Emerging Standard
Summarize a set of development tools for software systems. [Understanding]

Developed Standard
Use a set of development tools for software systems, such as requirements tracking, modeling, version control, automation, and testing. [Applying]

Highly Developed Standard
Choose a set of development tools for software systems. [Evaluating]

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

Emerging Standard
Paraphrase the requirements for a key feature for a secure software system. Understanding]

Developed Standard
Implement the requirements for a secure software system. [Applying]

Highly Developed Standard
Develop the requirements for a secure software system. [Creating]

SE-08 Illustrate principles of secure software design [Applying]

Emerging Standard
Describe different software design principles. [Understanding]

Developed Standard
Illustrate principles of secure software design, such as least privilege, simplicity, separation of concerns, information hiding, coupling and cohesion, and code reuse. [Applying]

Highly Developed Standard
Create a program that employs secure software design principles. [Creating]
SE-09 Analyze an existing software design to improve its security. [Analyzing]

Emerging Standard
Identify possible stages of software design that may introduce a security vulnerability. [Remembering]

Developed Standard
Analyze an existing software design to improve its security. [Analyzing]

Highly Developed Standard
Debate whether a proposed solution/patch to the design can fix the vulnerability in a viable and effective way. [Evaluating]

SE-10 Describe the cost and tradeoffs associated with designing security into software. [Understanding]

Emerging Standard
Recognize situations where security designs are effectively applied in software. [Remembering]

Developed Standard
Describe the cost and tradeoffs associated with designing security into software. [Understanding]

Highly Developed Standard
Compare security software designs and associated costs and tradeoffs. [Analyzing]

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

Emerging Standard
Demonstrate a defined coding standard in a small software project. [Understanding]

Developed Standard
Implement a small software project that uses a defined coding standard. [Applying]

Highly Developed Standard
Justify the reason for using a given coding standard. [Evaluating]

SE-12 Differentiate between program validation and verification. [Understanding]

Emerging Standard
Define software engineering terms verification and validation. [Remembering]

Developed Standard
Differentiate between program validation and verification. [Understanding]

Highly Developed Standard
Apply software validation and verification for a given piece of code. [Applying]

SE-13 Implement in code different types of testing, including security, unit testing, system testing, integration testing, and interface usability. [Applying]

Emerging Standard
Describe different types and levels of testing. [Understanding]

Developed Standard
Implement in code different types of testing, including security, unit testing, system testing, integration testing,
and interface usability. [Applying]
Highly Developed Standard
Examine different types of testing for given code. [Analyzing]

SF-01 Illustrate the basic building blocks of computers and their role in the historical development of computer architecture. [Applying]
Emerging Standard
Describe some of the basic building blocks of computers. [Understanding]
Developed Standard
Illustrate the basic building blocks of computers and their role in the historical development of computer architecture. [Applying]
Highly Developed Standard
Outline the basic building blocks of computers and their role in the historical development of computer architecture. [Analyzing]

SF-02 Discuss the differences between single thread and multiple thread, as well as single server and multiple server models. [Understanding]
Emerging Standard
Identify some differences between single thread and multiple thread, as well as single server and multiple server models. [Remembering]
Developed Standard
Discuss the differences between single thread and multiple thread, as well as single server and multiple server models. [Understanding]
Highly Developed Standard
Illustrate the differences between single thread and multiple thread, as well as single server and multiple server models. [Applying]

SF-03 Investigate security implications related to emerging computational paradigms. [Applying]
Emerging Standard
Report security implications related to computational paradigms. [Understanding]
Developed Standard
Investigate security implications related to emerging computational paradigms, such as quantum computing and biological computing. [Applying]

Highly Developed Standard
Analyze security implications related to emerging computational paradigms. [Analyzing]

**SF-04 Describe how computing systems are constructed of layers upon layers. [Understanding]**

Emerging Standard
Recognize that computing systems are constructed of layers upon layers. [Remembering]

Developed Standard
Describe how computing systems are constructed of layers upon layers, such as separation of concerns, well-defined interfaces, and abstraction. [Understanding]

Highly Developed Standard
Diagram a computing system constructed of layers upon layers. [Applying]

**SF-05 Implement a program using methods of layering. [Applying]**

Emerging Standard
Exemplify a program that uses methods of layering. [Understanding]

Developed Standard
Implement a program using methods of layering, such as error detection, recovery and status across layers. [Applying]

Highly Developed Standard
Develop a program using methods of layering. [Creating]

**SF-06 Investigate defects in a layered program using tools for program tracing, single stepping, and debugging. [Applying]**

Emerging Standard
Demonstrate defects in a layered program using tools for program tracing, single stepping, and debugging. [Understanding]

Developed Standard
Investigate defects in a layered program using tools for program tracing, single stepping, and debugging. [Applying]

Highly Developed Standard
Categorize defects by security risk in a layered program using tools for program tracing, single stepping, and debugging. [Analyzing]

**SF-07 Illustrate the performance of simple sequential and parallel versions of the same program with different problem sizes. [Applying]**
Emerging Standard
Summarize the general performance of simple sequential and parallel versions of the same program.  
[Understanding]
Developed Standard
Illustrate the performance of simple and parallel versions of the same program with different problem sizes.  
[Applying]
Highly Developed Standard
Compare the performance of simple and parallel versions of the same program with different problem sizes.  
[Analyzing]

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.

Emerging Standard
Define the concepts of instruction parallelism, data parallelism, thread parallelism/multitasking, and task/request parallelism. [Remembering]
Developed Standard
Summarize the differences among the concepts of instruction parallelism, data parallelism, thread parallelism/multitasking, and task/request parallelism. [Understanding]
Highly Developed Standard
Investigate the differences among the concepts of instruction parallelism, data parallelism, thread parallelism/multitasking, and task/request parallelism. [Applying]

SF-09 Investigate other uses of parallelism, including reliability and redundancy of execution. [Applying]

Emerging Standard
Describe other uses of parallelism, including reliability and redundancy of execution. [Understanding]
Developed Standard
Investigate other uses of parallelism, including reliability and redundancy of execution. [Applying]
Highly Developed Standard
Examine other uses of parallelism, including reliability and redundancy of execution. [Analyzing]

SP-01 Investigate both positive and negative ways in which computing technology impacts information exchange and social interaction. [Applying]

Emerging Standard
Describe different ways in which computing technology impacts information exchange and social interaction. [Understanding]
Developed Standard
Investigate both positive and negative ways in which computing technology impacts information exchange and social interaction, such as the Internet, mobile computing, and social media. [Applying]
Highly Developed Standard
Analyze positive and negative ways in which computing technology impacts information exchange and social interactions. [Analyzing]
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.

Emerging Standard
Infer developers’ assumptions and values embedded in hardware and software design, especially with respect to underrepresented groups and diverse populations. [Understanding]

Developed Standard
Examine developers’ assumptions and values embedded in hardware and software design, especially with respect to underrepresented groups and diverse populations, such as persons with disabilities. [Analyzing]

Highly Developed Standard
Critique developers’ assumptions and values embedded in hardware and software design, especially as pertinent to underrepresented groups and the disabled. [Evaluating]

SP-03 Analyze the impact of diversity on the computing profession. [Analyzing]

Emerging Standard
Discuss the impact of diversity on the computing profession. [Understanding]

Developed Standard
Analyze the impact of diversity on the computing profession, such as industry culture and product development. [Analyzing]

Highly Developed Standard
Assess the impact of diversity on the computing profession. [Evaluating]

SP-04 Investigate social engineering attacks and the types of bad actors who might perform them. [Applying]

Emerging Standard
Describe social engineering attacks and the types of bad actors who might perform them. [Understanding]

Developed Standard
Investigate social engineering attacks and the types of bad actors who might perform them. [Applying]

Highly Developed Standard
Analyze the impact and likelihood of social engineering attacks. [Analyzing]

SP-05 Contrast stakeholder positions in a given scenario. [Analyzing]

Emerging Standard
Infer stakeholder positions in a given scenario. [Understanding]

Developed Standard
Contrast stakeholder positions in a given scenario. [Analyzing]

Highly Developed Standard
Debate stakeholder positions in a given scenario. [Evaluating]
SP-06 Analyze social tradeoffs in technical decisions. [Analyzing]

Emerging Standard
Explain social tradeoffs in technical decisions. [Understanding]

Developed Standard
Analyze ethical and social tradeoffs in technical decisions. [Analyzing]

Highly Developed Standard
Justify social tradeoffs in technical decisions. [Evaluating]

SP-07 Examine various ethics scenarios in computing. [Analyzing]

Emerging Standard
Discuss ethics scenarios in computing. [Understanding]

Developed Standard
Analyze various ethics scenarios in computing. [Analyzing]

Highly Developed Standard
Debate various ethics scenarios in computing. [Evaluating]

SP-08 Support the ethical responsibility of ensuring software correctness, reliability, and safety. [Evaluating]

Emerging Standard
Examine the ethical responsibility in ensuring software correctness, reliability, and safety. [Analyzing]

Developed Standard
Support the ethical responsibility of ensuring software correctness, reliability, and safety. [Evaluating]

Highly Developed Standard
Hypothesize various ethical responsibilities of ensuring software correctness, reliability, and safety. [Creating]

SP-09 Compare professional codes of conduct from the ACM, IEEE Computer Society, and other organizations. [Analyzing]

Emerging Standard
Discuss professional codes of conduct from the ACM, IEEE Computer Society, and other organizations. [Understanding]

Developed Standard
Compare professional codes of conduct from the ACM, IEEE Computer Society, and other organizations. [Analyzing]

Highly Developed Standard
Evaluate professional codes of conduct from the ACM, IEEE Computer Society, and other organizations. [Evaluating]
SP-10 Differentiate the terms among intellectual property, fair-use, copyright, patent, trademark, and plagiarism. [Understanding]

Emerging Standard
Define the terms intellectual property, fair-use, copyright, and plagiarism. Give examples of each. State the plagiarism policy at your school. [Remembering]

Developed Standard
Differentiate the terms intellectual property, fair-use, copyright, patent, trademark, and plagiarism. [Understanding]

Highly Developed Standard
Investigate ethics violations related to intellectual property rights, fair-use, copyright, patents, trademarks, and plagiarism. [Applying]

SP-11 Discuss the rationale for legal protection of intellectual property. [Understanding]

Emerging Standard
Recognize the rationale for legal protection of intellectual property. [Remembering]

Developed Standard
Discuss the rationale for legal protection of intellectual property. [Understanding]

Highly Developed Standard
Examine the rationale for legal protection of intellectual property. [Analyzing]

SP-12 Outline the need for legal protection of personal privacy. [Analyzing]

Emerging Standard
Discuss the need for legal protection of personal privacy. [Understanding]

Developed Standard
Outline the need for legal protection of personal privacy. [Analyzing]

Highly Developed Standard
Defend the need for legal protection of personal privacy. [Evaluating]

SP-13 Investigate threats to privacy rights in personally identifiable information (PII). [Applying]

Emerging Standard
Summarize threats to privacy rights in personally identifiable information (PII). [Understanding]

Developed Standard
Investigate threats to privacy rights in personally identifiable information (PII). [Applying]

Highly Developed Standard
Analyze solutions for privacy threats to personally identifiable Information. [Analyzing]

SP-14 Illustrate the role of data collection in the implementation of pervasive surveillance systems. [Applying]

Emerging Standard
Discuss the role of data collection in the implementation of pervasive surveillance systems. [Understanding]
Developed Standard
Illustrate the role of data collection in the implementation of pervasive surveillance systems, such as RFID, face recognition, and mobile computing. [Applying]
Highly Developed Standard
Assess the role of data collection in the implementation of pervasive surveillance systems. [Evaluating]

**SP-15 Analyze technological solutions to privacy concerns. [Analyzing]**

Emerging Standard
Investigate technological solutions to privacy concerns. [Applying]
Developed Standard
Analyze technological solutions to privacy concerns. [Analyzing]
Highly Developed Standard
Choose a technological solution to solve a privacy problem. [Evaluating]

**SP-16 Use effective oral, written, electronic, and visual communication techniques with stakeholders. [Applying]**

Emerging Standard
Demonstrate effective oral, written, electronic, and visual communication techniques. [Understanding]
Developed Standard
Use effective oral, written, electronic, and visual communication techniques with stakeholders. [Applying]
Highly Developed Standard
Choose the appropriate oral, written, electronic or visual communication technique with stakeholders. [Evaluating]

**SP-17 Interpret the impact of both verbal and nonverbal cues during communication among team members. [Understanding]**

Emerging Standard
Recognize both verbal and nonverbal cues during communication among team members. [Remembering]
Developed Standard
Interpret the impact of both verbal and nonverbal cues during communication among team members. [Understanding]
Highly Developed Standard
Analyze the impact of both verbal and nonverbal cues during communication among team members. [Analyzing]

**SP-18 Develop technical artifacts. [Creating]**

Emerging Standard
Write a technical artifact. [Applying]
Developed Standard
Develop technical artifacts, such as documentation of source code and user requirements, as well as project
Highly Developed Standard
Create technical artifacts of considerable length and/or complexity. [Creating]

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

Emerging Standard
Paraphrase case studies related to sustainable computing efforts. [Understanding]
Developed Standard
Analyze case studies related to sustainable computing efforts. [Analyzing]
Highly Developed Standard
Critique case studies related to sustainable computing efforts. [Evaluating]

**SP-20 Investigate laws applicable to computer crimes. [Applying]**

Emerging Standard
Paraphrase laws applicable to computer crimes. [Understanding]
Developed Standard
Investigate laws applicable to computer crimes. [Applying]
Highly Developed Standard
Debate laws applicable to computer crimes. [Evaluating]

**SP-21 Examine the motivation and ramifications of cyber terrorism and criminal hacking. [Analyzing]**

Emerging Standard
Discuss the motivation and ramifications of cyber terrorism and criminal hacking. [Understanding]
Developed Standard
Examine the motivation and ramifications of cyber terrorism and criminal hacking. [Analyzing]
Highly Developed Standard
Evaluate the motivation and ramifications of cyber terrorism and criminal hacking. [Evaluating]

**SP-22 Write a company-wide security policy. [Applying]**

Emerging Standard
Exemplify a company-wide security policy. [Understanding]
Developed Standard
Write a company-wide security policy, such as procedures for managing passwords, avoiding social engineering attacks, and monitoring employees. [Applying]
Highly Developed Standard
Compare several company-wide security policies. [Analyzing]