Level 3A: GRADES 9-10		
Computing Systems (CS.3A) Conceptual understanding: People interact with a wide variety of computing devices that collect, store, analyze, and act upon information in ways that can affect human capabilities both positively and negatively. The physical components (hardware) and instructions (software) that make up a computing system communicate and process information in digital form. An understanding of hardware and software is useful when troubleshooting a computing system that does not work as intended.		
ID	**Standard**	**500167 GD1**
CS.3A.1	Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. [DEVICES] (P4.1) Computing devices are often integrated with other systems, including biological, mechanical, and social systems. A medical device can be embedded inside a person to monitor and regulate his or her health, a hearing aid (a type of assistive device) can filter out certain frequencies and magnify others, a monitoring device installed in a motor vehicle can track a person's driving patterns and habits, and a facial recognition device can be integrated into a security system to identify a person. The creation of integrated or embedded systems is not an expectation at this level.	Unit 7
CS.3A.1a	Students should be able to identify embedded computer systems.	
CS.31.1b	Students should describe the types of data and procedures that are included in the embedded system and explain how the implementation details are hidden from the user. For example, a students might select a car stereo, identify the types of data (radio station presets, station name or number, volume level) and procedures (increase volume, store/recall saved station, mute) it includes.	
CS.3A.2	Compare levels of abstraction and interactions between application software, system software, and hardware layers. [HARDWARE & SOFTWARE] (P4.1) At its most basic level, a computer is composed of physical hardware and electrical impulses. Multiple layers of software are built upon the hardware and interact with the layers above and below them to reduce complexity. System software manages a computing device's resources so that software can interact with hardware. System software is used on many different types of devices, such as smart TVs, assistive devices, virtual components, cloud components, and drones. For example, students may explore the progression from voltage to binary signal to logic gates to adders and so on. Knowledge of specific, advanced terms for computer architecture, such as BIOS, kernel, or bus, is not expected at this level	Unit 7
CS.3A.2a	Students should be able to distinguish between hardware and software	
CS.3A.2b	Students should be able to describe the purpose of and differences between system software (i.e. operating system) and application software (i.e. word processor).	
CS.3A.2c	Students should be able to describe how software and hardware interact. For example, text-editing software interacts with the operating system to receive input from the keyboard, convert the input to bits for storage, and interpret the bits as readable text to display on the monitor.	
CS.3A.3	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. [TROUBLESHOOTING] (P6.2) Troubleshooting complex problems involves the use of multiple sources when researching, evaluating, and implementing potential solutions. Troubleshooting also relies on experience, such as when people recognize that a problem is similar to one they have seen before or adapt solutions that have worked in the past. Examples of complex troubleshooting strategies include resolving connectivity problems, adjusting system configurations and settings, and ensuring hardware and software compatibility, and transferring data from one device to another	
CS.3A.3a	Students should develop guidelines by creating an artifact that conveys systematic troubleshooting strategies (i.e. create a flow chart or a job aid for a help desk employee).	

Level 3A: GRADES 9-10 - Networks and the Internet		
ID	**Networks and the Internet (NI.3A)**	**500167 GD1**
	Conceptual understanding: Computing devices typically do not operate in isolation. Networks connect computing devices to share information and resources and are an increasingly integral part of computing. Networks and communication systems provide greater connectivity in the computing world by providing fast, secure communication and facilitating innovation.	
NI.3A.1	Evaluate the scalability and reliability of networks by describing the relationship between routers, switches, servers, topology, and addressing. [NETWORK COMMUNICATION & ORGANIZATION] (P4.1) Each device is assigned an address that uniquely identifies it on the network. Routers function by comparing IP addresses to determine the pathways packets should take to reach their destination. Switches function by comparing MAC addresses to determine which computers or network segments will receive frames. Students could use online network simulators to experiment with these factors	
NI.3A.1a	Students should be able to define a MAC address – what is it and how it is used.	
NI.3A.1b	Students should be able to explain what a router and a switch are and how they work inside a network.	
NI.3A.1c	Students should be able to define what a server is and how it is used in a network	
NI.3A.1d	Students should be able to list various types of network topology and explain why each is used.	
NI.3A.1e	Students should be able to verbally and visually explain how addressing, routers, switches, and servers all work together in a network.	

NI.3A.2	Give examples to illustrate how sensitive data can be affected by malware and other attacks. [CYBERSECURITY] (P7.2) Network security depends on a combination of hardware, software, and practices that control access to data and systems. The needs of users and the sensitivity of data determine the level of security implemented. Potential security problems, such as denial-of-service attacks, ransomware, viruses, worms, spyware, and phishing, present threats to sensitive data.	
NI.3A.2a	Students should be able to discuss how sensitive data can be affected by malware and other attacks. Students might reflect on case studies or current events in which governments or organizations experienced data leaks or data loss as a result of these types of attacks	
NI.3A.3	Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. [CYBERSECURITY] (P3.1, 3.3) Security measures may include physical security tokens, two-factor authentication, and biometric verification. Potential security problems, such as denial-of-service attacks, ransomware, viruses, worms, spyware, and phishing, exemplify why sensitive data should be MISSISSIPPI COLLEGE- and CAREER-READINESS STANDARDS for COMPUTER SCIENCE 2018 40 Level 3A: Grades 9-10 securely stored and transmitted. The timely and reliable access to data and information services by authorized users, referred to as availability, is ensured through adequate bandwidth, backups, and other measures	
NI.3A.3a	Students should understand the different types of security problems and the different types of devices that can be impacted. Potential security problems may include issues such as denial-of-service attacks, ransomware, viruses, worms, spyware, phishing, and social engineering. Some types of devices impacted may include laptops, tablets, cell phones, self-driving cars, ATMs, and others	
NI.3A.3b	Students should systematically evaluate different security measures based on efficiency, feasibility, and ethical impacts. Students might address issues such as how efficiency affects feasibility or whether a proposed approach raises ethical concerns.	
NI.3A.4	Compare various security measures considering tradeoffs between the usability and security of a computing system. [CYBERSECURITY] (P6.3) Security measures may include physical security tokens, two-factor authentication, and biometric verification, but choosing security measures involves tradeoffs between the usability and security of the system. The needs of users and the sensitivity of data determine the level of security implemented.	
NI.3A.4a	Students should be able to explain different types of security measures and discuss the tradeoffs between usability and security. For example, students might discuss computer security policies in place at the local level that present a tradeoff between usability and security, such as a web filter that prevents access to many educational sites but keeps the campus network safe.	
NI.3A.5	Explain tradeoffs when selecting and implementing cybersecurity recommendations. [CYBERSECURITY] (P7.2) Network security depends on a combination of hardware, software, and practices that control access to data and systems. The needs of users and the sensitivity of data determine the level of security implemented. Every security measure involves tradeoffs between the accessibility and security of the system.	
NI.3A.5a	Students should be able to describe, justify, and document choices they make using terminology appropriate for the intended audience and purpose. Students could debate issues from the perspective of diverse audiences, including individuals, corporations, privacy advocates, security experts, and government.	

Level 3A: GRADES 9-10 - Data and Analysis

ID	Data and Analysis (DA.3A)	500167 GD1
	Conceptual understanding: Computing systems exist to process data. The amount of digital data generated in the world is rapidly expanding, so the need to process data effectively is increasingly important. Data is collected and stored so that it can be analyzed to better understand the world and make more accurate predictions.	
DA.3A.1	Translate between different bit representations of real-world phenomena, such as characters, numbers, and images. [STORAGE] (P4.1)	Unit 2
DA.3A.1a	Students should be able to translate between different bit representations. For example, convert hexadecimal color codes to decimal percentages, ASCII/Unicode representation, or converting binary to base 10.	
DA.3A.1b	Students should be able to discuss how data sequences can be interpreted in a variety of formats. For example, text, numbers, sound, and images	
DA.3A.2	Evaluate the tradeoffs in how data elements are organized and where data is stored. [STORAGE] (P3.3) People make choices about how data elements are organized and where data is stored. These choices affect cost, speed, reliability, accessibility, privacy, and integrity.	Unit 2, Unit 7
DA.3A.2a	Students should evaluate whether a chosen solution is most appropriate for a particular problem. Students might consider the cost, speed, reliability, accessibility, privacy, and integrity tradeoffs between storing photo data on a mobile device versus in the cloud.	
DA.3A.3	Collect, transform, and organize data to help others better understand a problem. [COLLECTION, VISUALIZATION, & TRANSFORMATION] (P4.4) People transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information. Examples include visualization, aggregation, rearrangement, and application of mathematical operations. People use software tools or programming to create powerful, interactive data visualizations and perform a range of mathematical operations to transform and analyze data.	
DA.3A.3a	Students should use various data collection techniques for different types of computational problems. For example, user surveys, mobile device GPS, social media data sets, etc.	
DA.3A.3b	Use computational tools to collect, transform, and organize data to help others better understand a problem.	

DA.3A.3c	Students should use data analysis to identify significant patterns in data sets.	
DA.3A.4	Create and evaluate computational models that represent real-world systems. [INFERENCE & MODELS] (P4.4) Computational models make predictions about processes or phenomenon based on selected data and features. The amount, quality, and diversity of data and the features chosen can affect the quality of a model and ability to understand a system. Predictions or inferences are tested to validate models.	
DA.3A.4a	Students should create computational models that simulate real-world systems (e.g., ecosystems, epidemics, spread of disease).	Unit 1, 2, 4, 5, 6, 7
DA.3A.4b	Students should analyze and evaluate the ability of models and simulations to formulate, refine, and test hypothesis.	

Level 3A: GRADES 9-10 - Algorithms and Programming

ID	Algorithms and Programming (AP.3A)	500167 GD1
	Conceptual understanding: An algorithm is a sequence of steps designed to accomplish a specific task. Algorithms are translated into programs, or code, to provide instructions for computing devices. Algorithms and programming control all computing systems, empowering people to communicate with the world in new ways and solve compelling problems. The development process to create meaningful and efficient programs involves choosing which information to use and how to process and store it, breaking apart large problems into smaller ones, recombining existing solutions, and analyzing different solutions.	Unit 2, 4, 5, 6, 7
AP.3A.1	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. [ALGORITHMS] (P5.2) A prototype is a computational artifact that demonstrates the core functionality of a product or process. Prototypes are useful for getting early feedback in the design process and can yield insight into the feasibility of a product. The process of developing computational artifacts MISSISSIPPI COLLEGE- and CAREER-READINESS STANDARDS for COMPUTER SCIENCE 2018 42 Level 3A: Grades 9-10 embraces both creative expression and the exploration of ideas to create prototypes and solve computational problems	Unit 1, 2, 4, 5, 6, 7
AP.3A.1a	Students create artifacts that are personally relevant or beneficial to their community and beyond. Students should develop artifacts in response to a task or a computational problem that demonstrate the performance, reusability, and ease of implementation of an algorithm.	Unit 2, 4, 5, 6, 7
AP.3A.2	Use lists and functions to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. [VARIABLES] (P4.1)	Unit 2, 4, 5, 6, 7
AP.3A.2a	Students should be able to identify common features in multiple segments of code and substitute a single segment that uses lists (arrays) or functions to account for the differences.	Unit 2, 4, 5, 6, 7
AP.3A.3	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. [CONTROL] (P5.2) Implementation includes the choice of programming language, which affects the time and effort required to create a program. Readability refers to how clear the program is to other programmers and can be improved through documentation. The discussion of performance is limited to a theoretical understanding of execution time and storage requirements; a quantitative analysis is not expected. Control structures at this level may include conditional statements, loops, event handlers, and recursion	Unit 2
AP.3A.3a	Students should be able to justify by explaining the benefits and drawbacks of the selection of specific control structures with regard to implementation, readability, and program performance. For example, students might compare the readability and program performance of iterative and recursive implementations of procedures that calculate the Fibonacci sequence.	Unit 2
AP.3A.4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. [CONTROL] (P5.2) In this context, relevant computational artifacts include programs, mobile apps, or web apps. Events can be user-initiated, such as a button press, or system-initiated, such as a timer firing. At previous levels, students have learned to create and call procedures. Here, students design procedures that are called by events	Unit 2, Unit 7
AP.3A.4a	Students will design procedures that are called by events. Students might create a mobile app that updates a list of nearby points of interest when the device detects that its location has been changed.	Unit 2, 6 7
AP.3A.5	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. [MODULARITY] (P3.2)	Unit 2, 4, 5, 7
AP.3A.5a	Students should decompose complex problems into manageable subproblems that could potentially be solved with programs or procedures that already exist. For example, students could create an app to solve a community problem by connecting to an online database through an application programming interface (API).	Unit 2, 4, 5, 7
AP.3A.6	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. [MODULARITY] (P5.2) Computational artifacts can be created by combining and modifying existing artifacts or by developing new artifacts. Examples of computational artifacts include programs, simulations, visualizations, digital animations, robotic systems, and apps. Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall MISSISSIPPI COLLEGE- and CAREER-READINESS STANDARDS for COMPUTER SCIENCE 2018 43 Level 3A: Grades 9-10 purpose. Modules allow for better management of complex tasks. The focus at this level is understanding a program as a system with relationships between modules.	Unit 2, 4, 7

AP.3A.6a	Students will create artifacts by using procedures within a program, combinations of data, and procedures, or independent but interrelated programs. The choice of implementation, such as programming language or paradigm, may vary. Students could incorporate computer vision libraries to increase the capabilities of a robot or leverage open-source JavaScript libraries to expand the functionality of a web application	Unit 2, 4, 5, 6, 7
AP.3A.7	Systematically design and develop programs for broad audiences by incorporating feedback from users. [PROGRAM DEVELOPMENT] (P5.1) Examples of programs could include games, utilities, and mobile applications. Students at lower levels collect feedback and revise programs.	Unit 2, 4, 5, 6, 7, 8
AP.3A.1b	Students should do so through a systematic process that includes feedback from broad audiences. Students might create a user satisfaction survey and brainstorm distribution methods that could yield feedback from a diverse audience, documenting the process they took to incorporate selected feedback in product revisions	Unit 8
AP.3A.8	Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries. [PROGRAM DEVELOPMENT] (P7.3) Examples of software licenses include copyright, freeware, and the many open-source licensing schemes. At previous levels, students adhered to licensing schemes	Unit 8
AP.3A.8a	Students should consider licensing implications for their own work, especially when incorporating libraries and other resources. Students might consider two software libraries that address a similar need, justifying their choice based on the library that has the least restrictive license.	Unit 8
AP.3A.9	Evaluate and refine computational artifacts to make them more usable and accessible. [PROGRAM DEVELOPMENT] (P6.3) Testing and refinement is the deliberate and iterative process of improving a computational artifact. This process includes debugging (identifying and fixing errors) and comparing actual outcomes to intended outcomes.	
AP.3A.9a	Students should respond to the changing needs and expectations of end users and improve the performance, reliability, usability, and accessibility of artifacts. For example, students could incorporate feedback from a variety of end users to help guide the size and placement of menus and buttons in a user interface.	
AP.3A.10	Design and develop computational artifacts working in team roles using collaborative tools. [PROGRAM DEVELOPMENT] (P2.4) Collaborative tools could be as complex as source code version control system or as simple as a collaborative word processor. Team roles in pair programming are driver and navigator but could be more specialized in larger teams. As programs grow more complex, the choice of resources that aid program development becomes increasingly important and should be made by the students.	Unit 8
AP.3A.10a	Students will work in teams using collaborative tools to design and develop computational artifacts. Students might work as a team to develop a mobile application that addresses a problem relevant to the school or community, selecting appropriate tools to establish and manage the project timeline; design, share, and revise graphical user interface elements; and track planned, inprogress, and completed components.	Unit 8
AP.3A.11	Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs. [PROGRAM DEVELOPMENT] (P7.2) Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. These modules can be procedures within a program; combinations of data and procedures; or independent, but interrelated, programs. The development of complex programs is aided by resources such as libraries and tools to edit and manage parts of the program	Unit 8
AP.3A.11a	Students will document design decisions using text, graphics, presentations, and/or demonstrations.	Unit 8

Level 3A: GRADES 9-10 - Impacts of Computing

ID	Impacts of Computing (IC.3A)	500167 GD1
	Conceptual understanding: Computing affects many aspects of the world in both positive and negative ways at local, national, and global levels. Individuals and communities influence computing through their behaviors and cultural and social interactions, and in turn, computing influences new cultural practices. An informed and responsible person should understand the social implications of the digital world, including equity and access to computing.	
IC.3A.1	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. [CULTURE] (P1.2) Computing may improve, harm, or maintain practices. Equity deficits, such as minimal exposure to computing, access to education, and training opportunities, are related to larger, systemic problems in society.	
IC.3A.1a	Students should be able to evaluate the accessibility of a product to a broad group of end users, such as people who lack access to broadband or who have various disabilities.	
IC.3A.2b	Students should also begin to identify potential bias during the design process to maximize accessibility in product design	
IC.3A.2	Test and refine computational artifacts to reduce bias and equity deficits. [CULTURE] (P1.2) Biases could include incorrect assumptions developers have made about their user base. Equity deficits include minimal exposure to computing, access to education, and training opportunities.	
IC.3A.2a	Students should begin to identify potential bias during the design process to maximize accessibility in product design and become aware of professionally accepted accessibility standards to evaluate computational artifacts for accessibility.	
IC.3A.3	Demonstrate ways a given algorithm applies to problems across disciplines. [CULTURE] (P3.1) Computation can share features with disciplines, such as art and music, by algorithmically translating human intention into an artifact.	
IC.3A.3a	Students should be able to identify real-world problems that span multiple disciplines, such as increasing bike safety with new helmet technology, and that can be solved computationally.	

IC.3A.4	Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields. [SOCIAL INTERACTIONS] (P2.4) MISSISSIPPI COLLEGE- and CAREER-READINESS STANDARDS for COMPUTER SCIENCE 2018 45 Level 3A: Grades 9-10 Many aspects of society, especially careers, have been affected by the degree of communication afforded by computing. The increased connectivity between people in different cultures and in different career fields has changed the nature and content of many careers	
IC.3A.4a	Students should explore different collaborative tools and methods used to solicit input from team members, classmates, and others, such as participation in online forums or local communities. For example, students could compare ways different social media tools could help a team become more cohesive	
IC.3A.5	Explain the beneficial and harmful effects that intellectual property laws can have on innovation. [SAFETY, LAW, & ETHICS] (P7.3) Laws govern many aspects of computing, such as privacy, data, property, information, and identity. These laws can have beneficial and harmful effects, such as expediting or delaying advancements in computing and protecting or infringing upon people's rights. International differences in laws and ethics have implications for computing. For examples, laws that mandate the blocking of some file-sharing websites may reduce online piracy but can restrict the right to access information. Firewalls can be used to block harmful viruses and malware but can also be used for media censorship.	
IC.3A.5a	Students should be aware of intellectual property laws and be able to explain how they are used to protect the interests of innovators and how patent trolls abuse the laws for financial gain.	
IC.3A.6	Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users. [SAFETY, LAW, & ETHICS] (P7.2) Data can be collected and aggregated across millions of people, even when they are not actively engaging with or physically near the data collection devices. This automated and nonevident collection can raise privacy concerns, such as social media sites mining an account even when the user is not online. Other examples include surveillance video used in a store to track customers for security or information about purchase habits or the monitoring of road traffic to change signals in real time to improve road efficiency without drivers being aware. Methods and devices for collecting data can differ by the amount of storage required, level of detail collected, and sampling rates	
IC.3A.6a	Students should be able to explain the privacy concerns related to the collection and generation of data through automated processes	
IC.3A.7	Evaluate the social and economic implications of privacy in the context of safety, law, or ethics. [SAFETY, LAW, & ETHICS] (P7.3) Laws govern many aspects of computing, such as privacy, data, property, information, and identity. International differences in laws and ethics have implications for computing.	
IC.3A.7a	Students should evaluate the social and economic implications of privacy in the context of safety, law, or ethics. For example, students might review case studies or current events that present an ethical dilemma when an individual's right to privacy is at odds with the safety, security, or wellbeing of a community.	