Computer Science Principles Pacing Guide (w/ Standards)

Unit 1 - Digital Information		Standards
Week 1:	Welcome to CSP	CSTA K-12 Computer Science Standards (2017) IC - Impacts of Computing 3B-IC-27 - Predict how computational innovations that have revolutionized aspects of our culture might evolve.
	Representing Information	CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-21 - Evaluate and refine computational artifacts to make them more usable and accessible. DA - Data & Analysis 2-DA-07 - Represent data using multiple encoding schemes. 3A-DA-09 - Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.
	Circle Square Patterns	CSTA K-12 Computer Science Standards (2017) DA - Data & Analysis 3A-DA-09 - Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.
	Binary Numbers	CSP Conceptual Framework DAT-1 - The way that the computer represents data is different from the way that the data are interpreted and displayed for the user. Programs are used to translate data into a representation that is more easily understood by people. DAT-1.A - Explain how data can be represented using bits. DAT-1.A.2 - Computing devices represent data digitally, meaning that the lowest-level components of any value are bits.

		DAT-1.A.3 - Bit is shorthand for binary digit and is either 0 or 1. DAT-1.A.4 - A byte is 8 bits. DAT-1.C - For binary numbers: a. Calculate the binary (base 2) equivalent of a positive integer (base 10) and vice versa. b. Compare and order binary numbers. DAT-1.C.1 - Number bases, including binary and decimal, are used to represent data. DAT-1.C.2 - Binary (base 2) uses only combinations of the digits zero and one. DAT-1.C.3 - Decimal (base 10) uses only combinations of the digits 0 – 9. DAT-1.C.4 - As with decimal, a digit's position in the binary sequence determines its numeric value. The numeric value is equal to the bit's value (0 or 1) multiplied by the place value of its position. DAT-1.C.5 - The place value of each position is determined by the base raised to the power of the position. Positions are numbered starting at the rightmost position with 0 and increasing by 1 for each subsequent position to the left. CSTA K-12 Computer Science Standards (2017) DA - Data & Analysis 3A-DA-09 - Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.
	Overflow and Rounding	CSP Conceptual Framework DAT-1 - The way that the computer represents data is different from the way that the data are interpreted and displayed for the user. Programs are used to translate data into a representation that is more easily understood by people. DAT-1.B - Explain the consequences of using bits to represent data. DAT-1.B.1 - In many programming languages, integers are represented by a fixed number of bits, which limits the range of integer values and mathematical

		operations on those values. This limitation can result in overflow or other errors. DAT-1.B.2 - Other programming languages provide an abstraction through which the size of representable integers is limited only by the size of the computer's memory; this is the case for the language defined in the exam reference sheet. DAT-1.B.3 - In programming languages, the fixed number of bits used to represent real numbers limits the range and mathematical operations on these values; this limitation can result in round-off and other errors. Some real numbers are represented as approximations in computer storage. CSTA K-12 Computer Science Standards (2017) DA - Data & Analysis 3A-DA-09 - Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.
Week 2:	Representing Text	CSP Conceptual Framework DAT-1 - The way that the computer represents data is different from the way that the data are interpreted and displayed for the user. Programs are used to translate data into a representation that is more easily understood by people. DAT-1.A - Explain how data can be represented using bits. DAT-1.A.5 - Abstraction is the process of reducing complexity by focusing on the main idea. By hiding details irrelevant to the question at hand and bringing together related and useful details, abstraction reduces complexity and allows one to focus on the idea. DAT-1.A.6 - Bits are grouped to represent abstractions. These abstractions include, but are not limited to, numbers, characters, and color. DAT-1.A.7 - The same sequence of bits may represent different types of data in different contexts. CSTA K-12 Computer Science Standards (2017)

		AP - Algorithms & Programming 3A-AP-21 - Evaluate and refine computational artifacts to make them more usable and accessible. DA - Data & Analysis 3A-DA-09 - Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.
	Black and White Images	CSP Conceptual Framework DAT-1 - The way that the computer represents data is different from the way that the data are interpreted and displayed for the user. Programs are used to translate data into a representation that is more easily understood by people. DAT-1.A - Explain how data can be represented using bits. DAT-1.A.10 - Analog data can be closely approximated digitally using a sampling technique, which means measuring values of the analog signal at regular intervals called samples. The samples are measured to figure out the exact bits required to store each sample. DAT-1.A.7 - The same sequence of bits may represent different types of data in different contexts. DAT-1.A.8 - Analog data have values that change smoothly, rather than in discrete intervals, over time. Some examples of analog data include pitch and volume of music, colors of a painting, or position of a sprinter during a race. CSTA K-12 Computer Science Standards (2017) CS - Computing Systems 3A-CS-02 - Compare levels of abstraction and interactions between application software, system software and hardware layers. DA - Data & Analysis 2-DA-07 - Represent data using multiple encoding schemes.

		3A-DA-09 - Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.
	Color Images	CSP Conceptual Framework DAT-1 - The way that the computer represents data is different from the way that the data are interpreted and displayed for the user. Programs are used to translate data into a representation that is more easily understood by people. DAT-1.A - Explain how data can be represented using bits. DAT-1.A.9 - The use of digital data to approximate real-world analog data is an example of abstraction. CSTA K-12 Computer Science Standards (2017) CS - Computing Systems 3A-CS-02 - Compare levels of abstraction and interactions between application software, system software and hardware layers. DA - Data & Analysis 2-DA-07 - Represent data using multiple encoding schemes. 3A-DA-09 - Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.
	Lossless Compression	CSP Conceptual Framework DAT-1 - The way that the computer represents data is different from the way that the data are interpreted and displayed for the user. Programs are used to translate data into a representation that is more easily understood by people. DAT-1.D - Compare data compression algorithms to determine which is best in a particular context. DAT-1.D.1 - Data compression can reduce the size (number of bits) of transmitted or stored data. DAT-1.D.2 - Fewer bits does not necessarily mean less information.

		DAT-1.D.3 - The amount of size reduction from compression depends on both the amount of redundancy in the original data representation and the compression algorithm applied. DAT-1.D.4 - Lossless data compression algorithms can usually reduce the number of bits stored or transmitted while guaranteeing complete reconstruction of the original data. CSTA K-12 Computer Science Standards (2017) DA - Data & Analysis 3A-DA-10 - Evaluate the tradeoffs in how data elements are organized and where data is stored.
	Lossy Compression	CSP Conceptual Framework DAT-1 - The way that the computer represents data is different from the way that the data are interpreted and displayed for the user. Programs are used to translate data into a representation that is more easily understood by people. DAT-1.D - Compare data compression algorithms to determine which is best in a particular context. DAT-1.D.5 - Lossy data compression algorithms can significantly reduce the number of bits stored or transmitted but only allow reconstruction of an approximation of the original data. DAT-1.D.6 - Lossy data compression algorithms can usually reduce the number of bits stored or transmitted more than lossless compression algorithms. DAT-1.D.7 - In situations where quality or ability to reconstruct the original is maximally important, lossless compression algorithms are typically chosen. DAT-1.D.8 - In situations where minimizing data size or transmission time is maximally important, lossy compression algorithms are typically chosen. CSTA K-12 Computer Science Standards (2017) DA - Data & Analysis 3A-DA-10 - Evaluate the tradeoffs in how data elements are organized and where data is stored.

Week 3:	Intellectual Property	CSP Conceptual Framework IOC-1 - While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences. IOC-1.F - Explain how the use of computing could raise legal and ethical concerns. IOC-1.F.1 - Material created on a computer is the intellectual property of the creator or an organization. IOC-1.F.2 - Ease of access and distribution of digitized information raises intellectual property concerns regarding ownership, value, and use. IOC-1.F.3 - Measures should be taken to safeguard intellectual property. IOC-1.F.4 - The use of material created by someone else without permission and presented as one's own is plagiarism and may have legal consequences. IOC-1.F.5 - Some examples of legal ways to use materials created by someone else include:● Creative Commons—a public copyright license that enables the free distribution of an otherwise copyrighted work. This is used when the content creator wants to give others the right to share, use, and build upon the work they have created. ● open source—programs that are made freely available and may be redistributed and modified ● open access—online research output free of any and all restrictions on access and free of many restrictions on use, such as copyright or license restrictions IOC-1.F.6 - The use of material created by someone other than you should always be cited. IOC-1.F.7 - Creative commons, open source, and open access have enabled broad access to digital information. CSTA K-12 Computer Science Standards (2017) IC - Impacts of Computing 3A-IC-28 - Explain the beneficial and harmful effects that intellectual property laws can have on innovation.

	Project - Digital Information Dilemmas - Parts 1-2	CSTA K-12 Computer Science Standards (2017) IC - Impacts of Computing 2-IC-20 - Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. 3A-IC-24 - Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. 3A-IC-28 - Explain the beneficial and harmful effects that intellectual property laws can have on innovation.
	Assessment Day	
Week 4		
Unit 2 - The Internet		
	Welcome to the Internet	CSTA K-12 Computer Science Standards (2017) NI - Networks & the Internet 2-NI-04 - Model the role of protocols in transmitting data across networks and the Internet. 3A-NI-04 - Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. 3B-NI-03 - Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).
	Building a Network	CSP Conceptual Framework CSN-1 - Computer systems and networks facilitate how data is transferred. CSN-1.A - Explain how computing devices work together in a network. CSN-1.A.1 - A computing device is a physical artifact that can run a program. Some examples include computers, tablets, servers, routers, and smart sensors. CSN-1.A.2 - A computing system is a group of computing devices and programs working together for a common purpose. CSN-1.A.3 - A computer network is a group of interconnected computing devices capable of sending or receiving data.

		CSN-1.A.4 - A computer network is a type of a computing system. CSN-1.A.5 - A path between two computing devices on a computer network (a sender and a receiver) is a sequence of directly connected computing devices that begins at the sender and ends at the receiver. CSN-1.A.6 - Routing is the process of finding a path from sender to receiver. CSN-1.A.7 - The bandwidth of a computer network is the maximum amount of data that can be sent in a fixed amount of time. CSN-1.A.8 - Bandwidth is usually measured in bits per second. CSTA K-12 Computer Science Standards (2017) NI - Networks & the Internet 2-NI-04 - Model the role of protocols in transmitting data across networks and the Internet. 3A-NI-04 - Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. 3B-NI-03 - Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).
	The Need for Addressing	CSP Conceptual Framework CSN-1 - Computer systems and networks facilitate how data is transferred. CSN-1.B - Explain how the Internet works. CSN-1.B.1 - The Internet is a computer network consisting of interconnected networks that use standardized, open (nonproprietary) communication protocols. CSN-1.B.2 - Access to the Internet depends on the ability to connect a computing device to an Internet-connected device. CSN-1.B.3 - A protocol is an agreed-upon set of rules that specify the behavior of a system.

		CSN-1.B.4 - The protocols used in the Internet are open, which allows users to easily connect additional computing devices to the Internet. CSTA K-12 Computer Science Standards (2017) NI - Networks & the Internet 2-NI-04 - Model the role of protocols in transmitting data across networks and the Internet. 3A-NI-04 - Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.
	Routers and Redundancy	CSP Conceptual Framework CSN-1 - Computer systems and networks facilitate how data is transferred. CSN-1.B - Explain how the Internet works. CSN-1.B.5 - Routing on the Internet is usually dynamic; it is not specified in advance. CSN-1.E - For fault-tolerant systems, like the Internet: a. Describe the benefits of fault-tolerance b. Explain how a given system is fault-tolerant c. Identify vulnerabilities in a system. CSN-1.E.1 - The Internet has been engineered to be fault-tolerant, with abstractions for routing and transmitting data. CSN-1.E.2 - Redundancy is the inclusion of extra components that can be used to mitigate failure of a system if other components fail. CSN-1.E.3 - One way to accomplish network redundancy is by having more than one path between any two connected devices. CSN-1.E.4 - If a particular device or connection on the Internet fails, subsequent data will be sent via a different route, if possible. CSN-1.E.5 - When a system can support failures and still continue to function, it is called fault-tolerant. This is important because elements of complex systems fail at unexpected times, often in groups, and fault tolerance allows users to continue to use the network.

		CSN-1.E.6 - Redundancy within a system often requires additional resources but can provide the benefit of fault tolerance. CSN-1.E.7 - The redundancy of routing options between two points increases the reliability of the Internet and helps it scale to more devices and more people. CSTA K-12 Computer Science Standards (2017) NI - Networks & the Internet 2-NI-04 - Model the role of protocols in transmitting data across networks and the Internet. 3A-NI-04 - Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. 3B-NI-03 - Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).
	Packets	CSP Conceptual Framework CSN-1 - Computer systems and networks facilitate how data is transferred. CSN-1.B - Explain how the Internet works. CSN-1.B.5 - Routing on the Internet is usually dynamic; it is not specified in advance. CSN-1.B.6 - The scalability of a system is the capacity for the system to change in size and scale to meet new demands. CSN-1.C - Explain how data is sent through the Internet via packets. CSN-1.C.1 - Information is passed through the Internet as a data stream. Data streams contain chunks of data, which are encapsulated in packets. CSN-1.C.2 - Packets contain a chunk of data and metadata used for routing the packet between the origin and the destination on the Internet, as well as for data reassembly. CSN-1.C.3 - Packets may arrive at the destination in order, out of order, or not at all. CSN-1.C.4 - IP, TCP, and UDP are common protocols used on the Internet.

		CSTA K-12 Computer Science Standards (2017) NI - Networks & the Internet 2-NI-04 - Model the role of protocols in transmitting data across networks and the Internet. 3A-NI-04 - Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. 3B-NI-03 - Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).
	HTTP and DNS	CSP Conceptual Framework CSN-1 - Computer systems and networks facilitate how data is transferred. CSN-1.B - Explain how the Internet works. CSN-1.B.7 - The Internet was designed to be scalable. CSN-1.D - Describe the differences between the Internet and the World Wide Web. CSN-1.D.1 - The World Wide Web is a system of linked pages, programs, and files. CSN-1.D.2 - HTTP is a protocol used by the World Wide Web. CSN-1.D.3 - The World Wide Web uses the Internet. IOC-2 - The use of computing innovations may involve risks to your personal safety and identity. IOC-2.B - Explain how computing resources can be protected and can be misused. IOC-2.B.6 - Certificate authorities issue digital certificates that validate the ownership of encryption keys used in secure communications and are based on a trust model . CSTA K-12 Computer Science Standards (2017) NI - Networks & the Internet 2-NI-04 - Model the role of protocols in transmitting data across networks and the Internet. 3A-NI-04 - Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.

		3B-NI-03 - Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).
Week 5:	Project - Internet Dilemmas - Parts 1-2	CSP Conceptual Framework IOC-1 - While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences. IOC-1.C - Describe issues that contribute to the digital divide. IOC-1.C.1 - Internet access varies between socioeconomic, geographic, and demographic characteristics, as well as between countries. IOC-1.C.2 - The “digital divide” refers to differing access to computing devices and the Internet, based on socioeconomic, geographic, or demographic characteristics. IOC-1.C.3 - The digital divide can affect both groups and individuals. IOC-1.C.4 - The digital divide raises issues of equity, access, and influence, both globally and locally. IOC-1.C.5 - The digital divide is affected by the actions of individuals, organizations, and governments. IOC-1.F - Explain how the use of computing could raise legal and ethical concerns. IOC-1.F.10 - The digital divide raises ethical concerns around computing. CSTA K-12 Computer Science Standards (2017) IC - Impacts of Computing 3A-IC-24 - Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. 3A-IC-28 - Explain the beneficial and harmful effects that intellectual property laws can have on innovation. 3A-IC-30 - Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.

		3B-IC-26 - Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society. 3B-IC-28 - Debate laws and regulations that impact the development and use of software.
	Assessment Day	
Unit 3 - Intro to App Design		
	Intro to Apps	CSP Conceptual Framework CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking. CRD-2.A - Describe the purpose of a computing innovation. CRD-2.A.2 - An understanding of the purpose of a computing innovation provides developers with an improved ability to develop that computing innovation. CRD-2.C - Identify input(s) to a program. CRD-2.C.1 - Program input is data sent to a computer for processing by a program. Input can come in a variety of forms, such as tactile, audio, visual, or text. CRD-2.C.4 - Inputs usually affect the output produced by a program. CRD-2.C.6 - Input can come from a user or other programs. CRD-2.D - Identify output(s) produced by a program. CRD-2.D.1 - Program output is any data sent from a program to a device. Program output can come in a variety of forms, such as tactile, audio, visual, or text. CRD-2.D.2 - Program output is usually based on a program's input or prior state (e.g., internal values).
Week 6:	Introduction to Design Mode	CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-16 - Design and iteratively develop computational artifacts for practical intent, personal

		expression, or to address a societal issue by using events to initiate instructions.
	Project - Designing an App - Parts 1-2	CSP Conceptual Framework CRD-1 - Incorporating multiple perspectives through collaboration improves the computing innovations being developed. CRD-1.A - Explain how computing innovations are improved through collaboration. CRD-1.A.3 - Effective collaboration produces a computing innovation that reflects the diversity of talents and perspectives of those who designed it. CRD-1.A.4 - Collaboration that includes diverse perspectives helps avoid bias in the development of computing innovations. CRD-1.A.6 - Information gathered from potential users can be used to understand the purpose of a program from diverse perspectives and to develop a program that fully incorporates these perspectives. CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking. CRD-2.E - Develop a program using a development process. CRD-2.E.1 - A development process can be ordered and intentional, or exploratory in nature. CRD-2.E.2 - There are multiple development processes. The following phases are commonly used when developing a program: ● investigating and reflecting ● designing ● prototyping ● testing CRD-2.E.4 - A development process that is incremental is one that breaks the problem into smaller pieces and makes sure each piece works before adding it to the whole. CRD-2.F - Design a program and its user interface. CRD-2.F.1 - The design of a program incorporates investigation to determine its requirements.

		CRD-2.F.2 - Investigation in a development process is useful for understanding and identifying the program constraints, as well as the concerns and interests of the people who will use the program. CRD-2.F.3 - Some ways investigation can be performed are as follows: ● collecting data through surveys ● user testing ● interviews ● direct observations CRD-2.F.4 - Program requirements describe how a program functions and may include a description of user interactions that a program must provide. CRD-2.F.5 - A program's specification defines the requirements for the program. CRD-2.F.6 - In a development process, the design phase outlines how to accomplish a given program specification. CRD-2.F.7 - The design phase of a program may include: ● brainstorming ● planning and storyboarding ● organizing the program into modules and functional components ● creation of diagrams that represent the layouts of the user interface ● CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-16 - Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. 3A-AP-19 - Systematically design and develop programs for broad audiences by incorporating feedback from users. 3A-AP-21 - Evaluate and refine computational artifacts to make them more usable and accessible. 3A-AP-22 - Design and develop computational artifacts working in team roles using collaborative tools. 3A-AP-23 - Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
	The Need for Programming languages	CSTA K-12 Computer Science Standards (2017)

		AP - Algorithms & Programming 2-AP-17 - Systematically test and refine programs using a range of test cases. 3A-AP-13 - Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. 3A-AP-21 - Evaluate and refine computational artifacts to make them more usable and accessible. 3A-AP-22 - Design and develop computational artifacts working in team roles using collaborative tools. 3A-AP-23 - Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
	Intro to Programming	CSP Conceptual Framework AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.B - Represent a step-by-step algorithmic process using sequential code statements. AAP-2.B.2 - A code statement is a part of program code that expresses an action to be carried out. AAP-2.B.6 - Sequential statements execute in the order they appear in the code segment. AAP-3 - Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing programmers to write programs more quickly and with more confidence. AAP-3.E - For generating random values: a. Write expressions to generate possible values. b. Evaluate expressions to determine the possible results.

		AAP-3.E.2 - Using random number generation in a program means each execution may produce a different result. CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking. CRD-2.B - Explain how a program or code segment functions. CRD-2.B.1 - A program is a collection of program statements that performs a specific task when run by a computer. A program is often referred to as software. CRD-2.B.3 - A program needs to work for a variety of inputs and situations. CRD-2.B.4 - The behavior of a program is how a program functions during execution and is often described by how a user interacts with it. CRD-2.B.5 - A program can be described broadly by what it does, or in more detail by both what the program does and how the program statements accomplish this function. CRD-2.C - Identify input(s) to a program. CRD-2.C.2 - An event is associated with an action and supplies input data to a program. CRD-2.C.3 - Events can be generated when a key is pressed, a mouse is clicked, a program is started, or by any other defined action that affects the flow of execution. CRD-2.C.5 - In event-driven programming, program statements are executed when triggered rather than through the sequential flow of control. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-16 - Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.

Week 7:	Debugging	CSP Conceptual Framework AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.B - Represent a step-by-step algorithmic process using sequential code statements. CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking. CRD-2.G - Describe the purpose of a code segment or program by writing documentation. CRD-2.G.1 - Program documentation is a written description of the function of a code segment, event, procedure, or program and how it was developed. CRD-2.G.2 - Comments are a form of program documentation written into the program to be read by people and do not affect how a program runs. CRD-2.G.3 - Programmers should document a program throughout its development. CRD-2.G.4 - Program documentation helps in developing and maintaining correct programs when working individually or in collaborative programming environments. CRD-2.G.5 - Not all programming environments support comments, so other methods of documentation may be required. CRD-2.I - For errors in an algorithm or program: a. Identify the error. b. Correct the error. CRD-2.I.5 - The following are effective ways to find and correct errors: ● test cases ● hand tracing ● visualizations ● debuggers ● adding extra output statement(s) CSTA K-12 Computer Science Standards (2017) CS - Computing Systems

		3A-CS-03 - Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
	Project - Designing an App - Parts 3-5	CSP Conceptual Framework CRD-1 - Incorporating multiple perspectives through collaboration improves the computing innovations being developed. CRD-1.A - Explain how computing innovations are improved through collaboration. CRD-1.A.5 - Consultation and communication with users are important aspects of the development of computing innovations. CRD-1.B - Explain how computing innovations are developed by groups of people. CRD-1.B.1 - Online tools support collaboration by allowing programmers to share and provide feedback on ideas and documents. CRD-1.B.2 - Common models such as pair programming exist to facilitate collaboration. CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking. CRD-2.E - Develop a program using a development process. CRD-2.E.3 - A development process that is iterative requires refinement and revision based on feedback, testing, or reflection throughout the process. This may require revisiting earlier phases of the process. CRD-2.J - Identify inputs and corresponding expected output or behaviors that can be used to check the correctness of an algorithm or program. CRD-2.J.1 - In the development process, testing uses defined inputs to ensure that an algorithm or program is producing the expected outcomes. Programmers use the results from testing to revise their algorithms or programs.

		CRD-2.J.3 - Program requirements are needed to identify appropriate defined inputs for testing. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-16 - Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. 3A-AP-19 - Systematically design and develop programs for broad audiences by incorporating feedback from users. 3A-AP-21 - Evaluate and refine computational artifacts to make them more usable and accessible. 3A-AP-22 - Design and develop computational artifacts working in team roles using collaborative tools. 3A-AP-23 - Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
	Assessment Day	
Unit 4 - Variables, Conditionals, and Functions		
	Variables Explore	CSP Conceptual Framework AAP-1 - To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways. AAP-1.A - Represent a value with a variable. AAP-1.A.1 - A variable is an abstraction inside a program that can hold a value. Each variable has associated data storage that represents one value at a time, but that value can be a list or other collection that in turn contains multiple values. AAP-1.A.2 - Using meaningful variable names helps with the readability of program code and understanding of what values are represented by the variables. AAP-1.A.3 - Some programming languages provide types to represent data, which are referenced using

		variables. These types include numbers, Booleans, lists, and strings. AAP-1.B - Determine the value of a variable as a result of an assignment. AAP-1.B.1 - The assignment operator allows a program to change the value represented by a variable. AAP-1.B.2 - The exam reference sheet provides the "←" operator to use for assignment. For example,a ← expression evaluates expression and then assigns the result to the variable a. AAP-1.B.3 - The value stored in a variable will be the most recent value assigned. For example: a ← 1b ← a a ← 2 display(b) still displays 1. AAP-1.C - Represent a list or string using a variable. AAP-1.C.4 - A string is an ordered sequence of characters. AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.B - Represent a step-by-step algorithmic process using sequential code statements. AAP-2.B.3 - An expression can consist of a value, a variable, an operator, or a procedure call that returns a value. AAP-2.B.4 - Expressions are evaluated to produce a single value. AAP-2.B.5 - The evaluation of expressions follows a set order of operations defined by the programming language. AAP-2.C - Evaluate expressions that use arithmetic operators. AAP-2.C.1 - Arithmetic operators are part of most programming languages and include addition, subtraction, multiplication, division, and modulus operators.

		AAP-2.C.2 - The exam reference sheet provides a MOD b, which evaluates to the remainder when a is divided by b. Assume that a is an integer greater than or equal to 0 and b is an integer greater than 0. For example, 17 MOD 5 evaluates to 2. AAP-2.C.3 - The exam reference sheet provides the arithmetic operators +, -, *, /, and MOD. ● a + b● a – b● a * b● a / b● a MOD bThese are used to perform arithmetic on a and b. For example, 17 / 5 evaluates to 3.4. AAP-2.C.4 - The order of operations used in mathematics applies when evaluating expressions. The MOD operator has the same precedence as the * and / operators. AAP-2.D - Evaluate expressions that manipulate strings. AAP-2.D.1 - String concatenation joins together two or more strings end-to-end to make a new string. AAP-3 - Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing programmers to write programs more quickly and with more confidence. AAP-3.A - For procedure calls: a. Write statements to call procedures. b. Determine the result or effect of a procedure call. AAP-3.A.6 - The exam reference sheet provides the procedure DISPLAY(expression) to display the value of expression, followed by a space. AAP-3.E - For generating random values: a. Write expressions to generate possible values. b. Evaluate expressions to determine the possible results. AAP-3.E.1 - The exam reference sheet provides RANDOM(a, b) which generates and returns a random integer from a to b, inclusive. Each result is equally

		likely to occur. For example, RANDOM(1, 3) could return 1, 2, or 3. DAT-1 - The way that the computer represents data is different from the way that the data are interpreted and displayed for the user. Programs are used to translate data into a representation that is more easily understood by people. DAT-1.A - Explain how data can be represented using bits. DAT-1.A.1 - Data values can be stored in variables, lists of items, or standalone constants and can be passed as input to (or output from) procedures. DAT-1.B - Explain the consequences of using bits to represent data. DAT-1.B.2 - Other programming languages provide an abstraction through which the size of representable integers is limited only by the size of the computer's memory; this is the case for the language defined in the exam reference sheet. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug. 3B-AP-23 - Evaluate key qualities of a program through a process such as a code review.
Week 8:	Variables Investigate	CSP Conceptual Framework AAP-1 - To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways. AAP-1.A - Represent a value with a variable. AAP-1.A.2 - Using meaningful variable names helps with the readability of program code and understanding of what values are represented by the variables.

		AAP-1.A.3 - Some programming languages provide types to represent data, which are referenced using variables. These types include numbers, Booleans, lists, and strings. AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.D - Evaluate expressions that manipulate strings. AAP-2.D.1 - String concatenation joins together two or more strings end-to-end to make a new string. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 3B-AP-23 - Evaluate key qualities of a program through a process such as a code review.
	Variables Practice	CSP Conceptual Framework AAP-1 - To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways. AAP-1.B - Determine the value of a variable as a result of an assignment. AAP-1.B.2 - The exam reference sheet provides the "←" operator to use for assignment. For example,a ← expression evaluates expression and then assigns the result to the variable a. AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.C - Evaluate expressions that use arithmetic operators.

		AAP-2.C.2 - The exam reference sheet provides a MOD b, which evaluates to the remainder when a is divided by b. Assume that a is an integer greater than or equal to 0 and b is an integer greater than 0. For example, 17 MOD 5 evaluates to 2. AAP-2.C.3 - The exam reference sheet provides the arithmetic operators +, -, *, /, and MOD. ● a + b● a – b● a * b● a / b● a MOD bThese are used to perform arithmetic on a and b. For example, 17 / 5 evaluates to 3.4. AAP-3 - Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing programmers to write programs more quickly and with more confidence. AAP-3.A - For procedure calls: a. Write statements to call procedures. b. Determine the result or effect of a procedure call. AAP-3.A.6 - The exam reference sheet provides the procedure DISPLAY(expression) to display the value of expression, followed by a space. AAP-3.E - For generating random values: a. Write expressions to generate possible values. b. Evaluate expressions to determine the possible results. AAP-3.E.1 - The exam reference sheet provides RANDOM(a, b) which generates and returns a random integer from a to b, inclusive. Each result is equally likely to occur. For example, RANDOM(1, 3) could return 1, 2, or 3. DAT-1 - The way that the computer represents data is different from the way that the data are interpreted and displayed for the user. Programs are used to translate data into a representation that is more easily understood by people.

		DAT-1.B - Explain the consequences of using bits to represent data. DAT-1.B.2 - Other programming languages provide an abstraction through which the size of representable integers is limited only by the size of the computer's memory; this is the case for the language defined in the exam reference sheet. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values.
	Variables Make	CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.
	Conditionals Explore	CSP Conceptual Framework AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.E - For relationships between two variables, expressions, or values: a. Represent using relational operators b. Evaluate expressions that use relational operators. AAP-2.E.1 - A Boolean value is either true or false. AAP-2.F - For relationships between Boolean values: a. Represent using logical operators. b. Evaluate expressions that use logic operators. AAP-2.F.5 - The operand for a logical operator is either a Boolean expression or a single Boolean value. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming

		2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 3A-AP-15 - Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance and explain the benefits and drawbacks of choices made.
	Conditionals Investigate	CSP Conceptual Framework AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.H - For selection: a. Represent using conditional statements. b. Determine the result of conditional statements. AAP-2.H.1 - Conditional statements or “if-statements” affect the sequential flow of control by executing different statements based on the value of a Boolean expression. AAP-2.I - For nested selection: a. Represent using nested conditional statements. b. Determine the result of nested conditional statements. AAP-2.I.1 - Nested conditional statements, or “else if” statements, consist of conditional statements within conditional statements. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 3B-AP-23 - Evaluate key qualities of a program through a process such as a code review.
Week 9:	Conditionals Practice	CSP Conceptual Framework AAP-2 - The way statements are sequenced and combined in a program determines the computed

		result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.E - For relationships between two variables, expressions, or values: a. Represent using relational operators b. Evaluate expressions that use relational operators. AAP-2.E.2 - The exam reference sheet provides the following relational operators: =, ≠, >, <, ≥, and ≤. ● a = b● a ≠ b● a > b● a < b● a ≥ b● a ≤ b These are used to test the relationship between two variables, expressions, or values. A comparison using a relational operator evaluates to a Boolean value. For example, a = b evaluates to true if a and b are equal; otherwise, it evaluates to false. AAP-2.F - For relationships between Boolean values: a. Represent using logical operators. b. Evaluate expressions that use logic operators. AAP-2.F.1 - The exam reference sheet provides the logical operators NOT, AND, and OR, which evaluate to a Boolean value. AAP-2.F.2 - The exam reference sheet provides the NOT condition, which evaluates to true if condition is false; otherwise it evaluates to false. AAP-2.F.3 - The exam reference sheet provides condition1 AND condition2, which evaluates to true if both condition1 and condition2 are true; otherwise it evaluates to false. AAP-2.F.4 - The exam reference sheet provides condition1 OR condition2, which evaluates to true if condition1 is true or if condition2 is true or if both condition1 and condition2 are true; otherwise it evaluates to false. AAP-2.H - For selection: a. Represent using conditional statements. b. Determine the result of conditional statements.

		AAP-2.H.2 - The exam reference sheet providesIF(condition){<block of statements>}in which the code in block of statements is executed if the Boolean expression condition evaluates to true; no action is taken if condition evaluates to false. AAP-2.H.3 - The exam reference sheet providesIF(condition){ <first block of statements>}ELSE{ <second block of statements>}in which the code in first block of statements is executed if the Boolean expression condition evaluates to true; otherwise, the code in second block of statements is executed. AAP-2.L - Compare multiple algorithms to determine if they yield the same side effect or result. AAP-2.L.3 - Some conditional statements can be written as equivalent Boolean expressions. AAP-2.L.4 - Some Boolean expressions can be written as equivalent conditional statements. AAP-3 - Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing programmers to write programs more quickly and with more confidence. AAP-3.A - For procedure calls: a. Write statements to call procedures. b. Determine the result or effect of a procedure call. AAP-3.A.9 - The exam reference sheet provides procedure INPUT(), which accepts a value from the user and returns the input value. CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking. CRD-2.I - For errors in an algorithm or program: a. Identify the error. b. Correct the error.

		CRD-2.I.1 - A logic error is a mistake in the algorithm or program that causes it to behave incorrectly or unexpectedly. CRD-2.I.2 - A syntax error is a mistake in the program where the rules of the programming language are not followed. CRD-2.I.3 - A run-time error is a mistake in the program that occurs during the execution of a program. Programming languages define their own run-time errors. CRD-2.I.4 - An overflow error is an error that occurs when a computer attempts to handle a number that is outside of the defined range of values. CRD-2.J - Identify inputs and corresponding expected output or behaviors that can be used to check the correctness of an algorithm or program. CRD-2.J.2 - Defined inputs used to test a program should demonstrate the different expected outcomes that are at or just beyond the extremes (minimum and maximum) of input data. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
	Conditionals Make	CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.

		3B-AP-21 - Develop and use a series of test cases to verify that a program performs according to its design specifications.
	Functions Explore/Investigate	CSP Conceptual Framework AAP-3 - Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing programmers to write programs more quickly and with more confidence. AAP-3.A - For procedure calls: a. Write statements to call procedures. b. Determine the result or effect of a procedure call. AAP-3.A.1 - A procedure is a named group of programming instructions that may have parameters and return values. AAP-3.A.2 - Procedures are referred to by different names, such as method or function, depending on the programming language. AAP-3.A.4 - A procedure call interrupts the sequential execution of statements, causing the program to execute the statements within the procedure before continuing. Once the last statement in the procedure (or a return statement) has executed, flow of control is returned to the point immediately following where the procedure was called. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-17 - Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. 3B-AP-23 - Evaluate key qualities of a program through a process such as a code review.
	Functions Practice	CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming

		3A-AP-17 - Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. 3B-AP-14 - Construct solutions to problems using student-created components, such as procedures, modules and/or objects.
	Functions Make	CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-19 - Document programs in order to make them easier to follow, test, and debug. 3A-AP-17 - Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. 3B-AP-14 - Construct solutions to problems using student-created components, such as procedures, modules and/or objects.
	Project - Decision Maker App - Parts 1 - 3	CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 3A-AP-16 - Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. 3B-AP-14 - Construct solutions to problems using student-created components, such as procedures, modules and/or objects.
	Assessment Day	
Unit 5 - Data		
Week 10	Learning from Data	CSP Conceptual Framework

		DAT-2 - Programs can be used to process data, which allows users to discover information and create new knowledge. DAT-2.A - Describe what information can be extracted from data. DAT-2.A.1 - Information is the collection of facts and patterns extracted from data. DAT-2.A.2 - Data provide opportunities for identifying trends, making connections, and addressing problems. DAT-2.A.3 - Digitally processed data may show correlation between variables. A correlation found in data does not necessarily indicate that a causal relationship exists. Additional research is needed to understand the exact nature of the relationship. DAT-2.A.4 - Often, a single source does not contain the data needed to draw a conclusion. It may be necessary to combine data from a variety of sources to formulate a conclusion. DAT-2.B - Describe what information can be extracted from metadata. DAT-2.B.1 - Metadata are data about data. For example, the piece of data may be an image, while the metadata may include the date of creation or the file size of the image. DAT-2.B.2 - Changes and deletions made to metadata do not change the primary data. DAT-2.B.3 - Metadata are used for finding, organizing, and managing information. DAT-2.B.4 - Metadata can increase the effective use of data or data sets by providing additional information. DAT-2.B.5 - Metadata allow data to be structured and organized. DAT-2.D - Extract information from data using a program. DAT-2.D.3 - Search tools are useful for efficiently finding information.

		DAT-2.E - Explain how programs can be used to gain insight and knowledge from data. DAT-2.E.1 - Programs are used in an iterative and interactive way when processing information to allow users to gain insight and knowledge about data. CSTA K-12 Computer Science Standards (2017) DA - Data & Analysis 3A-DA-11 - Create interactive data visualizations using software tools to help others better understand real-world phenomena. 3B-DA-05 - Use data analysis tools and techniques to identify patterns in data representing complex systems.
	Exploring One Column	CSP Conceptual Framework DAT-2 - Programs can be used to process data, which allows users to discover information and create new knowledge. DAT-2.D - Extract information from data using a program. DAT-2.D.1 - Programs can be used to process data to acquire information. DAT-2.D.2 - Tables, diagrams, text, and other visual tools can be used to communicate insight and knowledge gained from data. DAT-2.D.5 - Programs such as spreadsheets help efficiently organize and find trends in information. DAT-2.D.6 - Some processes that can be used to extract or modify information from data include the following:● transforming every element of a data set, such as doubling every element in a list, or extracting the parent's email from every student record● filtering a data set, such as keeping only the positive numbers from a list, or keeping only students who signed up for band from a record of all the students ● combining or comparing data in some way, such as adding up a list of numbers, or finding the student who has the highest GPA ● visualizing a

		data set through a chart, graph, or other visual representation CSTA K-12 Computer Science Standards (2017) DA - Data & Analysis 3A-DA-11 - Create interactive data visualizations using software tools to help others better understand real-world phenomena. 3B-DA-05 - Use data analysis tools and techniques to identify patterns in data representing complex systems. 3B-DA-06 - Select data collection tools and techniques to generate data sets that support a claim or communicate information.
	Filtering and Cleaning Data	CSP Conceptual Framework DAT-2 - Programs can be used to process data, which allows users to discover information and create new knowledge. DAT-2.C - Identify the challenges associated with processing data. DAT-2.C.1 - The ability to process data depends on the capabilities of the users and their tools. DAT-2.C.2 - Data sets pose challenges regardless of size, such as:● the need to clean data● incomplete data● invalid data● the need to combine data sources DAT-2.C.3 - Depending on how data were collected, they may not be uniform. For example, if users entered data into an open field, the way they choose to abbreviate, spell, or capitalize something may vary from user to user. DAT-2.C.4 - Cleaning data is a process that makes the data uniform without changing its meaning (e.g., replacing all equivalent abbreviations, spellings, and capitalizations with the same word). DAT-2.D - Extract information from data using a program.

		DAT-2.D.4 - Data filtering systems are important tools for finding information and recognizing patterns in data. DAT-2.E - Explain how programs can be used to gain insight and knowledge from data. DAT-2.E.2 - Programmers can use programs to filter and clean digital data, thereby gaining insight and knowledge. DAT-2.E.3 - Combining data sources, clustering data, and classifying data are parts of the process of using programs to gain insight and knowledge from data. CSTA K-12 Computer Science Standards (2017) DA - Data & Analysis 3A-DA-11 - Create interactive data visualizations using software tools to help others better understand real-world phenomena. 3B-DA-05 - Use data analysis tools and techniques to identify patterns in data representing complex systems. 3B-DA-06 - Select data collection tools and techniques to generate data sets that support a claim or communicate information.
	Exploring Two Columns	CSP Conceptual Framework DAT-2 - Programs can be used to process data, which allows users to discover information and create new knowledge. DAT-2.E - Explain how programs can be used to gain insight and knowledge from data. DAT-2.E.4 - Insight and knowledge can be obtained from translating and transforming digitally represented information. DAT-2.E.5 - Patterns can emerge when data are transformed using programs. CSTA K-12 Computer Science Standards (2017) DA - Data & Analysis 3A-DA-11 - Create interactive data visualizations using software tools to help others better understand real-world phenomena.

		3B-DA-05 - Use data analysis tools and techniques to identify patterns in data representing complex systems. 3B-DA-06 - Select data collection tools and techniques to generate data sets that support a claim or communicate information.
	Big, Open and Crowdsourced Data	CSP Conceptual Framework DAT-2 - Programs can be used to process data, which allows users to discover information and create new knowledge. DAT-2.C - Identify the challenges associated with processing data. DAT-2.C.6 - The size of a data set affects the amount of information that can be extracted from it. DAT-2.C.7 - Large data sets are difficult to process using a single computer and may require parallel systems. DAT-2.C.8 - Scalability of systems is an important consideration when working with data sets, as the computational capacity of a system affects how data sets can be processed and stored. IOC-1 - While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences. IOC-1.E - Explain how people participate in problem-solving processes at scale. IOC-1.E.1 - Widespread access to information and public data facilitates the identification of problems, development of solutions, and dissemination of results. IOC-1.E.2 - Science has been affected by using distributed and “citizen science” to solve scientific problems. IOC-1.E.3 - Citizen science is scientific research conducted in whole or part by distributed individuals, many of whom may not be scientists, who contribute relevant data to research using their own computing devices.

		IOC-1.E.4 - Crowdsourcing is the practice of obtaining input or information from a large number of people via the Internet. IOC-1.E.5 - Human capabilities can be enhanced by collaboration via computing. IOC-1.E.6 - Crowdsourcing offers new models for collaboration, such as connecting businesses or social causes with funding. CSTA K-12 Computer Science Standards (2017) DA - Data & Analysis 3A-DA-10 - Evaluate the tradeoffs in how data elements are organized and where data is stored.
	Machine Learning	CSP Conceptual Framework DAT-2 - Programs can be used to process data, which allows users to discover information and create new knowledge. DAT-2.C - Identify the challenges associated with processing data. DAT-2.C.5 - Problems of bias are often created by the type or source of data being collected. Bias is not eliminated by simply collecting more data. IOC-1 - While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences. IOC-1.B - Explain how a computing innovation can have an impact beyond its intended purpose. IOC-1.B.1 - Computing innovations can be used in ways that their creators had not originally intended:● The World Wide Web was originally intended only for rapid and easy exchange of information within the scientific community. ● Targeted advertising is used to help businesses, but it can be misused at both individual and aggregate levels. ● Machine learning and data mining have enabled innovation in medicine, business, and science, but information discovered in this way has also been used to discriminate against groups of individuals.

		IOC-1.B.3 - Responsible programmers try to consider the unintended ways their computing innovations can be used and the potential beneficial and harmful effects of these new uses. IOC-1.D - Explain how bias exists in computing innovations. IOC-1.D.1 - Computing innovations can reflect existing human biases because of biases written into the algorithms or biases in the data used by the innovation. IOC-1.D.2 - Programmers should take action to reduce bias in algorithms used for computing innovations as a way of combating existing human biases. IOC-1.D.3 - Biases can be embedded at all levels of software development. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3B-AP-08 - Describe how artificial intelligence drives many software and physical systems.
Week 11	Algorithmic Bias	CSP Conceptual Framework IOC-1 - While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences. IOC-1.B - Explain how a computing innovation can have an impact beyond its intended purpose. IOC-1.B.1 - Computing innovations can be used in ways that their creators had not originally intended:● The World Wide Web was originally intended only for rapid and easy exchange of information within the scientific community. ● Targeted advertising is used to help businesses, but it can be misused at both individual and aggregate levels. ● Machine learning and data mining have enabled innovation in medicine, business, and science, but information discovered in this way has also been used to discriminate against groups of individuals.

		IOC-1.B.2 - Some of the ways computing innovations can be used may have a harmful impact on society, economy, or culture. IOC-1.D - Explain how bias exists in computing innovations. IOC-1.D.1 - Computing innovations can reflect existing human biases because of biases written into the algorithms or biases in the data used by the innovation. IOC-1.D.2 - Programmers should take action to reduce bias in algorithms used for computing innovations as a way of combating existing human biases. IOC-1.D.3 - Biases can be embedded at all levels of software development. IOC-1.F - Explain how the use of computing could raise legal and ethical concerns. IOC-1.F.11 - Computing innovations can raise legal and ethical concerns. Some examples of these include:● the development of software that allows access to digital media downloads and streaming● the development of algorithms that include bias● the existence of computing devices that collect and analyze data by continuously monitoring activities
	Project-Tell a Data Story	CSTA K-12 Computer Science Standards (2017) DA - Data & Analysis 3A-DA-11 - Create interactive data visualizations using software tools to help others better understand real-world phenomena. 3B-DA-05 - Use data analysis tools and techniques to identify patterns in data representing complex systems. 3B-DA-06 - Select data collection tools and techniques to generate data sets that support a claim or communicate information.
	Assessment Day	
Unit 6 - Lists, Loops, and Traversals		
	Lists Explore	CSP Conceptual Framework

		AAP-1 - To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways. AAP-1.C - Represent a list or string using a variable. AAP-1.C.1 - A list is an ordered sequence of elements. For example, [value1, value2, value3, …] describes a list where value1 is the first element, value 2 is the second element, value 3 is the third element, and so on. AAP-1.C.2 - An element is an individual value in a list that is assigned a unique index. AAP-1.C.3 - An index is a common method for referencing the elements in a list or string using natural numbers. AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.N - For list operations: a. Write expressions that use list indexing and list procedures. b. Evaluate expressions that use list indexing and list procedures. AAP-2.N.2 - List procedures are implemented in accordance with the syntax rules of the programming language. AAP-2.O - For algorithms involving elements of a list: a. Represent using iterative statements to traverse a list. b. Determine the result of an algorithm with list traversals. AAP-2.O.3 - The exam reference sheet providesFOR EACH item IN aList{ <block of statements>}The variable item is assigned the value of each element of aList sequentially, in order, from the first element to the last element. The code in block of statements is executed once for each assignment of item. AAP-2.O.4 - Knowledge of existing algorithms that use iteration can help in constructing new algorithms.

		Some examples of existing algorithms that are often used with lists include:● determining a minimum or maximum value in a list● computing a sum or average of a list of numbers CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-14 - Use lists to simplify solutions, generalizing computational problems instead of repeated use of simple variables.
	Lists Investigate	CSP Conceptual Framework AAP-1 - To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways. AAP-1.A - Represent a value with a variable. AAP-1.A.3 - Some programming languages provide types to represent data, which are referenced using variables. These types include numbers, Booleans, lists, and strings. AAP-1.A.4 - Some values are better suited to representation using one type of data rather than another. AAP-1.D - For data abstraction: a. Develop data abstraction using lists to store multiple elements. b. Explain how the use of data abstraction manages complexity in program code. AAP-1.D.1 - Data abstraction provides a separation between the abstract properties of a data type and the concrete details of its representation. AAP-1.D.2 - Data abstractions manage complexity in programs by giving a collection of data a name without referencing the specific details of the representation. AAP-1.D.3 - Data abstractions can be created using lists. AAP-1.D.4 - Developing a data abstraction to implement in a program can result in a program that is easier to develop and maintain. AAP-1.D.5 - Data abstractions often contain different types of elements.

		AAP-1.D.6 - The use of lists allows multiple related items to be treated as a single value. Lists are referred to by different names, such as array, depending on the programming language. EXCLUSION STATEMENT(EK APP-1.D.6): The use of linked lists is outside the scope of this course and the AP Exam. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-14 - Use lists to simplify solutions, generalizing computational problems instead of repeated use of simple variables. 3B-AP-23 - Evaluate key qualities of a program through a process such as a code review.
	Lists Practice	CSP Conceptual Framework AAP-1 - To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways. AAP-1.D - For data abstraction: a. Develop data abstraction using lists to store multiple elements. b. Explain how the use of data abstraction manages complexity in program code. AAP-1.D.7 - The exam reference sheet provides the notation [value1, value2, value3…] to create a list with those values as the first, second, third, and so on items. For example, ● aList ← [value1, value2, value3,...] creates a new list that contains the values value1, value2, value3, and ... at indices 1, 2, 3, and ... respectively and assigns it to aList. aList ← [] creates a new empty list and assigns it to aList. AAP-1.D.8 - The exam reference sheet describes a list structure whose index values are 1 through the number of elements in the list, inclusive. For all list operations, if a list index is less than 1 or greater than the length of the list, an error message is produced and the program will terminate. AAP-2 - The way statements are sequenced and combined in a program determines the computed

		result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.D - Evaluate expressions that manipulate strings. AAP-2.D.2 - A substring is part of an existing string. AAP-2.N - For list operations: a. Write expressions that use list indexing and list procedures. b. Evaluate expressions that use list indexing and list procedures. AAP-2.N.1 - The exam reference sheet provides basic operations on lists, including: ● accessing an element by index ● aList[i] accesses the element of aList at index i. The first element of aList is at index 1 and accessed using the notation aList[1]. ● assigning a value of an element of a list to a variable x aList ← [i] assigns the value of aList[i] to the variable x. ● assigning a value to an element of a list aList[i] ← x assigns the value of x to aList[i]. AAP-2.N.2 - List procedures are implemented in accordance with the syntax rules of the programming language. AAP-2.O - For algorithms involving elements of a list: a. Represent using iterative statements to traverse a list. b. Determine the result of an algorithm with list traversals. AAP-2.O.3 - The exam reference sheet providesFOR EACH item IN aList{ <block of statements>}The variable item is assigned the value of each element of aList sequentially, in order, from the first element to the last element. The code in block of statements is executed once for each assignment of item. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-14 - Use lists to simplify solutions, generalizing computational problems instead of repeated use of simple variables.
Week 12	Lists Make	CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming

		3A-AP-14 - Use lists to simplify solutions, generalizing computational problems instead of repeated use of simple variables.
	Loops Explore	CSP Conceptual Framework AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.K - For iteration: a. Represent using iteration statements. b. Determine the result or side-effect of iteration statements. AAP-2.K.1 - Iteration statements change the sequential flow of control by repeating a set of statements zero or more times, until a stopping condition is met. AAP-2.K.4 - In REPEAT UNTIL(condition) iteration, an infinite loop occurs when the ending condition will never evaluate to true. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-15 - Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance and explain the benefits and drawbacks of choices made.
	Loops Investigate	CSP Conceptual Framework AAP-3 - Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing programmers to write programs more quickly and with more confidence. AAP-3.F - For simulations: a. Explain how computers can be used to represent real-world phenomena or outcomes. b. Compare the use of simulations with real-world contexts.

		AAP-3.F.1 - Simulations are abstractions of more complex objects or phenomena for a specific purpose. AAP-3.F.2 - A simulation is a representation that uses varying sets of values to reflect the changing state of a phenomenon. AAP-3.F.3 - Simulations often mimic real-world events with the purpose of drawing inferences, allowing investigation of a phenomenon without the constraints of the real world. AAP-3.F.4 - The process of developing an abstract simulation involves removing specific details or simplifying functionality. AAP-3.F.5 - Simulations can contain bias derived from the choices of real-world elements that were included or excluded. AAP-3.F.6 - Simulations are most useful when real-world events are impractical for experiments (e.g., too big, too small, too fast, too slow, too expensive, or too dangerous). AAP-3.F.7 - Simulations facilitate the formulation and refinement of hypotheses related to the objects or phenomena under consideration. AAP-3.F.8 - Random number generators can be used to simulate the variability that exists in the real world. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-15 - Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance and explain the benefits and drawbacks of choices made. 3B-AP-23 - Evaluate key qualities of a program through a process such as a code review. DA - Data & Analysis 3A-DA-12 - Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.
	Loops Practice	CSP Conceptual Framework

		AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.K - For iteration: a. Represent using iteration statements. b. Determine the result or side-effect of iteration statements. AAP-2.K.2 - The exam reference sheet providesREPEAT n TIMES{ <block of statements>}in which the block of statements is executed n times. AAP-2.K.3 - The exam reference sheet providesREPEAT UNTIL(condition){ <block of statements>}in which the code in block of statements is repeated until the Boolean expression condition evaluates to true. AAP-2.K.4 - In REPEAT UNTIL(condition) iteration, an infinite loop occurs when the ending condition will never evaluate to true. AAP-2.K.5 - In REPEAT UNTIL(condition) iteration, if the conditional evaluates to true initially, the loop body is not executed at all, due to the condition being checked before the loop. AAP-2.L - Compare multiple algorithms to determine if they yield the same side effect or result. AAP-2.L.1 - Algorithms can be written in different ways and still accomplish the same tasks. AAP-2.L.2 - Algorithms that appear similar can yield different side effects or results. AAP-2.L.5 - Different algorithms can be developed or used to solve the same problem. CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking. CRD-2.I - For errors in an algorithm or program: a. Identify the error. b. Correct the error.

		CRD-2.I.3 - A run-time error is a mistake in the program that occurs during the execution of a program. Programming languages define their own run-time errors. CRD-2.I.4 - An overflow error is an error that occurs when a computer attempts to handle a number that is outside of the defined range of values. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-15 - Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance and explain the benefits and drawbacks of choices made. DA - Data & Analysis 3A-DA-12 - Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.
	Loops Make	CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-15 - Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance and explain the benefits and drawbacks of choices made.
	Traversals Explore	CSP Conceptual Framework AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.O - For algorithms involving elements of a list: a. Represent using iterative statements to traverse a list. b. Determine the result of an algorithm with list traversals. AAP-2.O.1 - Traversing a list can be a complete traversal, where all elements in the list are accessed, or a partial traversal, where only a portion of elements are accessed. EXCLUSION STATEMENT (EK AAP-2.O.1):

		Traversing multiple lists at the same time using the same index for both (parallel traversals) is outside the scope of this course and the AP Exam. AAP-2.O.2 - Iteration statements can be used to traverse a list. AAP-2.O.4 - Knowledge of existing algorithms that use iteration can help in constructing new algorithms. Some examples of existing algorithms that are often used with lists include:● determining a minimum or maximum value in a list● computing a sum or average of a list of numbers DAT-2 - Programs can be used to process data, which allows users to discover information and create new knowledge. DAT-2.D - Extract information from data using a program. DAT-2.D.6 - Some processes that can be used to extract or modify information from data include the following:● transforming every element of a data set, such as doubling every element in a list, or extracting the parent's email from every student record● filtering a data set, such as keeping only the positive numbers from a list, or keeping only students who signed up for band from a record of all the students ● combining or comparing data in some way, such as adding up a list of numbers, or finding the student who has the highest GPA ● visualizing a data set through a chart, graph, or other visual representation CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-14 - Use lists to simplify solutions, generalizing computational problems instead of repeated use of simple variables. 3B-AP-10 - Use and adapt classic algorithms to solve computational problems.
	Traversals Investigate	CSP Conceptual Framework

		AAP-3 - Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing programmers to write programs more quickly and with more confidence. AAP-3.F - For simulations: a. Explain how computers can be used to represent real-world phenomena or outcomes. b. Compare the use of simulations with real-world contexts. AAP-3.F.1 - Simulations are abstractions of more complex objects or phenomena for a specific purpose. AAP-3.F.2 - A simulation is a representation that uses varying sets of values to reflect the changing state of a phenomenon. AAP-3.F.3 - Simulations often mimic real-world events with the purpose of drawing inferences, allowing investigation of a phenomenon without the constraints of the real world. AAP-3.F.4 - The process of developing an abstract simulation involves removing specific details or simplifying functionality. AAP-3.F.5 - Simulations can contain bias derived from the choices of real-world elements that were included or excluded. AAP-3.F.6 - Simulations are most useful when real-world events are impractical for experiments (e.g., too big, too small, too fast, too slow, too expensive, or too dangerous). AAP-3.F.7 - Simulations facilitate the formulation and refinement of hypotheses related to the objects or phenomena under consideration. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming

		3A-AP-14 - Use lists to simplify solutions, generalizing computational problems instead of repeated use of simple variables. 3B-AP-10 - Use and adapt classic algorithms to solve computational problems. 3B-AP-23 - Evaluate key qualities of a program through a process such as a code review.
Week 13	Traversals Practice	CSP Conceptual Framework AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.O - For algorithms involving elements of a list: a. Represent using iterative statements to traverse a list. b. Determine the result of an algorithm with list traversals. AAP-2.O.3 - The exam reference sheet providesFOR EACH item IN aList{ <block of statements>}The variable item is assigned the value of each element of aList sequentially, in order, from the first element to the last element. The code in block of statements is executed once for each assignment of item. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-14 - Use lists to simplify solutions, generalizing computational problems instead of repeated use of simple variables. 3B-AP-10 - Use and adapt classic algorithms to solve computational problems.
	Traversals Make	CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-14 - Use lists to simplify solutions, generalizing computational problems instead of repeated use of simple variables. 3B-AP-10 - Use and adapt classic algorithms to solve computational problems.

	Project - Hackathon - Parts 1-3	CSP Conceptual Framework CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking. CRD-2.B - Explain how a program or code segment functions. CRD-2.B.2 - A code segment refers to a collection of program statements that are part of a program. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3A-AP-14 - Use lists to simplify solutions, generalizing computational problems instead of repeated use of simple variables. 3A-AP-16 - Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. 3A-AP-22 - Design and develop computational artifacts working in team roles using collaborative tools. 3A-AP-23 - Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
	Project - Hackathon - Parts 4-5	Same as Above
	Assessment Day	
Unit 7 - Parameters, Return, and Libraries		
	Parameters and Return Explore	CSP Conceptual Framework AAP-3 - Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing programmers to write programs more quickly and with more confidence.

		AAP-3.A - For procedure calls: a. Write statements to call procedures. b. Determine the result or effect of a procedure call. AAP-3.A.1 - A procedure is a named group of programming instructions that may have parameters and return values. AAP-3.A.3 - Parameters are input variables of a procedure. Arguments specify the values of the parameters when a procedure is called. AAP-3.B - For procedure calls: a. Write statements to call procedures. b. Determine the result or effect of a procedure call. AAP-3.B.4 - A procedural abstraction may extract shared features to generalize functionality instead of duplicating code. This allows for program code reuse, which helps manage complexity. AAP-3.B.5 - Using parameters allows procedures to be generalized, enabling the procedures to be reused with a range of input values or arguments. AAP-3.B.6 - Using procedural abstraction helps improve code readability. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-14 - Create procedures with parameters to organize code and make it easier to reuse.
	Parameters and Return Investigate	CSP Conceptual Framework AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.C - Evaluate expressions that use arithmetic operators. AAP-2.C.3 - The exam reference sheet provides the arithmetic operators +, -, *, /, and MOD. ● a + b● a – b● a * b● a / b● a MOD bThese are used

		to perform arithmetic on a and b. For example, 17 / 5 evaluates to 3.4. AAP-3 - Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing programmers to write programs more quickly and with more confidence. AAP-3.B - For procedure calls: a. Write statements to call procedures. b. Determine the result or effect of a procedure call. AAP-3.B.4 - A procedural abstraction may extract shared features to generalize functionality instead of duplicating code. This allows for program code reuse, which helps manage complexity. AAP-3.B.5 - Using parameters allows procedures to be generalized, enabling the procedures to be reused with a range of input values or arguments. AAP-3.B.6 - Using procedural abstraction helps improve code readability. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-14 - Create procedures with parameters to organize code and make it easier to reuse. 3B-AP-23 - Evaluate key qualities of a program through a process such as a code review.
Week 14:	Parameters and Return Practice	CSP Conceptual Framework AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.C - Evaluate expressions that use arithmetic operators. AAP-2.C.2 - The exam reference sheet provides a MOD b, which evaluates to the remainder when a is divided

		by b. Assume that a is an integer greater than or equal to 0 and b is an integer greater than 0. For example, 17 MOD 5 evaluates to 2. AAP-2.O - For algorithms involving elements of a list: a. Represent using iterative statements to traverse a list. b. Determine the result of an algorithm with list traversals. AAP-2.O.1 - Traversing a list can be a complete traversal, where all elements in the list are accessed, or a partial traversal, where only a portion of elements are accessed. EXCLUSION STATEMENT (EK AAP-2.O.1): Traversing multiple lists at the same time using the same index for both (parallel traversals) is outside the scope of this course and the AP Exam. AAP-3 - Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing programmers to write programs more quickly and with more confidence. AAP-3.A - For procedure calls: a. Write statements to call procedures. b. Determine the result or effect of a procedure call. AAP-3.A.5 - The exam reference sheet provides procName (arg1, arg2, …) as a way to call PROCEDURE procName(parameter1, parameter 2, …) which takes zero or more arguments; arg1 is assigned to parameter1, arg2 is assigned to parameter2, and so on. AAP-3.A.7 - The exam reference sheet provides the RETURN(expression) statement, which is used to return the flow of control to the point where the procedure was called and to return the value of expression. AAP-3.A.8 - The exam reference sheet provides result ß procName(arg1, arg2, …) to assign to result the “value of the procedure” being returned by calling PROCEDURE procName(parameter1, parameter2, …)

		AAP-3.C - Develop procedural abstractions to manage complexity in a program by writing procedures. AAP-3.C.1 - The exam reference sheet provides PROCEDURE procName(parameter1, parameter2, …){<block of statements>}which is used to define a procedure that takes zero or more arguments. The procedure contains block of statements. AAP-3.C.2 - The exam reference sheet providesPROCEDURE procName(parameter1, parameter2, …){ <block of statements> RETURN(expression)} which is used to define a procedure that takes zero or more arguments. The procedure contains block of statements and returns the value of expression. The RETURN statement may appear at any point inside the procedure and causes an immediate return from the procedure back to the calling statement. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-14 - Create procedures with parameters to organize code and make it easier to reuse.
	Parameters and Return Make	CSP Conceptual Framework CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking. CRD-2.F - Design a program and its user interface. CRD-2.F.7 - The design phase of a program may include: ● brainstorming ● planning and storyboarding ● organizing the program into modules and functional components ● creation of diagrams that represent the layouts of the user interface ● CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-14 - Create procedures with parameters to organize code and make it easier to reuse.
	Libraries Explore	CSP Conceptual Framework

		AAP-3 - Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing programmers to write programs more quickly and with more confidence. AAP-3.D - Select appropriate libraries or existing code segments to use in creating new programs. AAP-3.D.1 - A software library contains procedures that may be used in creating new programs. AAP-3.D.2 - Existing code segments can come from internal or external sources, such as libraries or previously written code. AAP-3.D.3 - The use of libraries simplifies the task of creating complex programs. AAP-3.D.4 - Application program interfaces (APIs) are specifications for how the procedures in a library behave and can be used. AAP-3.D.5 - Documentation for an API/library is necessary in understanding the behavior(s) provided by the API/library and how to use them. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-14 - Create procedures with parameters to organize code and make it easier to reuse. 3B-AP-16 - Demonstrate code reuse by creating programming solutions using libraries and APIs.
	Libraries Investigate	CSP Conceptual Framework AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.M - For algorithms: a. Create algorithms. b. Combine and modify existing algorithms.

		AAP-2.M.1 - Algorithms can be created from an idea, by combining existing algorithms, or by modifying existing algorithms. AAP-2.M.2 - Knowledge of existing algorithms can help in constructing new ones. Some existing algorithms include:● determining the maximum or minimum value of 2 or more numbers ● computing the sum or average of 2 or more numbers ● identifying if an integer is or is not evenly divisible by another integer ● determining a robot's path through a maze AAP-2.M.3 - Using existing correct algorithms as building blocks for constructing another algorithm has benefits such as reducing development time, reducing testing, and simplifying the identification of errors. AAP-3 - Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing programmers to write programs more quickly and with more confidence. AAP-3.B - For procedure calls: a. Write statements to call procedures. b. Determine the result or effect of a procedure call. AAP-3.B.1 - One common type of abstraction is procedural abstraction, which provides a name for a process and allows a procedure to be used only knowing what it does, not how it does it. AAP-3.B.2 - Procedural abstraction allows a solution to a large problem to be based on the solution of smaller subproblems. This is accomplished by creating procedures to solve each of the subproblems. AAP-3.B.3 - The subdivision of a computer program into separate subprograms is called modularity. AAP-3.B.4 - A procedural abstraction may extract shared features to generalize functionality instead of

		duplicating code. This allows for program code reuse, which helps manage complexity. AAP-3.B.5 - Using parameters allows procedures to be generalized, enabling the procedures to be reused with a range of input values or arguments. AAP-3.B.6 - Using procedural abstraction helps improve code readability. AAP-3.B.7 - Using procedural abstraction in a program allows programmers to change the internals of the procedure (to make it faster, more efficient, use less storage, etc.) without needing to notify users of the change as long as what the procedure does is preserved. AAP-3.D - Select appropriate libraries or existing code segments to use in creating new programs. AAP-3.D.2 - Existing code segments can come from internal or external sources, such as libraries or previously written code. AAP-3.D.3 - The use of libraries simplifies the task of creating complex programs. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-14 - Create procedures with parameters to organize code and make it easier to reuse. 2-AP-17 - Systematically test and refine programs using a range of test cases. 3A-AP-18 - Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. 3B-AP-16 - Demonstrate code reuse by creating programming solutions using libraries and APIs. 3B-AP-23 - Evaluate key qualities of a program through a process such as a code review.
	Libraries Practice	CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-14 - Create procedures with parameters to organize code and make it easier to reuse.

		2-AP-17 - Systematically test and refine programs using a range of test cases. 3A-AP-18 - Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. 3B-AP-16 - Demonstrate code reuse by creating programming solutions using libraries and APIs.
	Project - Make a Library - Parts 1-3	CSP Conceptual Framework CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking. CRD-2.H - Acknowledge code segments used from other sources. CRD-2.H.1 - It is important to acknowledge any code segments that were developed collaboratively or by another source. CRD-2.H.2 - Acknowledgement of code segment(s) written by someone else and used in a program can be in the program documentation. The acknowledgement should include the origin or original author's name. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 2-AP-14 - Create procedures with parameters to organize code and make it easier to reuse. 3A-AP-18 - Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. 3B-AP-14 - Construct solutions to problems using student-created components, such as procedures, modules and/or objects. 3B-AP-16 - Demonstrate code reuse by creating programming solutions using libraries and APIs.
	Assessment Day	
Unit 8 - Cybersecurity and Global Impacts		
Week 15:	Project - Innovation Simulation - Parts 1-2	CSP Conceptual Framework

		CRD-1 - Incorporating multiple perspectives through collaboration improves the computing innovations being developed. CRD-1.A - Explain how computing innovations are improved through collaboration. CRD-1.A.1 - A computing innovation includes a program as an integral part of its function. CRD-1.A.2 - A computing innovation can be physical (e.g., self-driving car), non-physical computing software (e.g., picture editing software), or non-physical computing concepts (e.g., e-commerce). CRD-1.C - Demonstrate effective interpersonal skills during collaboration. CRD-1.C.1 - Effective collaborative teams practice interpersonal skills, including but not limited to: ● communication ● consensus building ● conflict resolution ● negotiation CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking. CRD-2.A - Describe the purpose of a computing innovation. CRD-2.A.1 - The purpose of computing innovations is to solve problems or to pursue interests through creative expression. IOC-1 - While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences. IOC-1.A - Explain how an effect of a computing innovation can be both beneficial and harmful. IOC-1.A.1 - People create computing innovations. IOC-1.A.2 - The way people complete tasks often changes to incorporate new computing innovations. IOC-1.A.5 - Advances in computing have generated and increased creativity in other fields, such as medicine, engineering, communications, and the arts.

		IOC-2 - The use of computing innovations may involve risks to your personal safety and identity. IOC-2.B - Explain how computing resources can be protected and can be misused. IOC-2.B.1 - Authentication measures protect devices and information from unauthorized access. Examples of authentication measures include strong passwords and multifactor authentication. IOC-2.B.10 - All real-world systems have errors or design flaws that can be exploited to compromise them. Regular software updates help fix errors that could compromise a computing system. IOC-2.B.2 - A strong password is something that is easy for a user to remember but would be difficult for someone else to guess based on knowledge of that user. IOC-2.B.3 - Multifactor authentication is a method of computer access control in which a user is only granted access after successfully presenting several separate pieces of evidence to an authentication mechanism, typically in at least two of the following categories: knowledge (something they know), possession (something they have), and inherence (something they are). IOC-2.B.4 - Multifactor authentication requires at least two steps to unlock protected information; each step adds a new layer of security that must be broken to gain unauthorized access. CSTA K-12 Computer Science Standards (2017) IC - Impacts of Computing 3A-IC-24 - Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. 3A-IC-27 - Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.

		3B-IC-25 - Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.
	Data Policies and Privacy	CSP Conceptual Framework IOC-2 - The use of computing innovations may involve risks to your personal safety and identity. IOC-2.A - Describe the risks to privacy from collecting and storing personal data on a computer system. IOC-2.A.1 - Personally identifiable information (PII) is information about an individual that identifies, links, relates, or describes them. Examples of PII include:● social security number● age● race● phone number(s)● medical information ● financial information ● biometric data IOC-2.A.12 - PII can be used to stalk or steal the identity of a person or to aid in the planning of other criminal acts. IOC-2.A.13 - Once information is placed online, it is difficult to delete. IOC-2.A.14 - Programs can collect your location and record where you have been, how you got there, and how long you were at a given location. IOC-2.A.15 - Information posted to social media services can be used by others. Combining information posted on social media and other sources can be used to deduce private information about you. IOC-2.A.2 - Search engines can record and maintain a history of searches made by users. IOC-2.A.3 - Websites can record and maintain a history of individuals who have viewed their pages. IOC-2.A.4 - Devices, websites, and networks can collect information about a user's location. IOC-2.A.5 - Technology enables the collection, use, and exploitation of information about, by, and for individuals, groups, and institutions. IOC-2.A.6 - Search engines can use search history to suggest websites or for targeted marketing.

		IOC-2.A.7 - Disparate personal data, such as geolocation, cookies, and browsing history, can be aggregated to create knowledge about an individual. IOC-2.A.8 - PII and other information placed online can be used to enhance a user's online experiences. IOC-2.A.9 - PII stored online can be used to simplify making online purchases. CSTA K-12 Computer Science Standards (2017) IC - Impacts of Computing 2-IC-23 - Describe tradeoffs between allowing information to be public and keeping information private and secure. 3A-IC-29 - Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users. 3A-IC-30 - Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.
	The Value of Privacy	CSP Conceptual Framework IOC-1 - While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences. IOC-1.A - Explain how an effect of a computing innovation can be both beneficial and harmful. IOC-1.A.3 - Not every effect of a computing innovation is anticipated in advance. IOC-1.B - Explain how a computing innovation can have an impact beyond its intended purpose. IOC-1.B.1 - Computing innovations can be used in ways that their creators had not originally intended:● The World Wide Web was originally intended only for rapid and easy exchange of information within the scientific community. ● Targeted advertising is used to help businesses, but it can be misused at both individual and aggregate levels. ● Machine learning and data mining have enabled innovation in medicine, business, and science, but information discovered in this way has

		also been used to discriminate against groups of individuals. IOC-1.F - Explain how the use of computing could raise legal and ethical concerns. IOC-1.F.11 - Computing innovations can raise legal and ethical concerns. Some examples of these include:● the development of software that allows access to digital media downloads and streaming● the development of algorithms that include bias● the existence of computing devices that collect and analyze data by continuously monitoring activities IOC-1.F.8 - As with any technology or medium, using computing to harm individuals or groups of people raises legal and ethical concerns. IOC-1.F.9 - Computing can play a role in social and political issues, which in turn often raise legal and ethical concerns. IOC-2 - The use of computing innovations may involve risks to your personal safety and identity. IOC-2.A - Describe the risks to privacy from collecting and storing personal data on a computer system. IOC-2.A.1 - Personally identifiable information (PII) is information about an individual that identifies, links, relates, or describes them. Examples of PII include:● social security number● age● race● phone number(s)● medical information ● financial information ● biometric data IOC-2.A.10 - Commercial and governmental curation of information may be exploited if privacy and other protections are ignored. IOC-2.A.11 - Information placed online can be used in ways that were not intended and that may have a harmful impact. For example, an email message may be forwarded, tweets can be retweeted, and social media posts can be viewed by potential employers.

		IOC-2.A.12 - PII can be used to stalk or steal the identity of a person or to aid in the planning of other criminal acts. IOC-2.A.15 - Information posted to social media services can be used by others. Combining information posted on social media and other sources can be used to deduce private information about you. CSTA K-12 Computer Science Standards (2017) IC - Impacts of Computing 2-IC-23 - Describe tradeoffs between allowing information to be public and keeping information private and secure. 3A-IC-29 - Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users. 3A-IC-30 - Evaluate the social and economic implications of privacy in the context of safety, law, or ethics. 3B-IC-28 - Debate laws and regulations that impact the development and use of software.
	Project - Innovation Simulation - Part 3	CSP Conceptual Framework IOC-1 - While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences. IOC-1.A - Explain how an effect of a computing innovation can be both beneficial and harmful. IOC-1.A.4 - A single effect can be viewed as both beneficial and harmful by different people, or even by the same person. CSTA K-12 Computer Science Standards (2017) IC - Impacts of Computing 3A-IC-24 - Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. 3A-IC-27 - Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.

		3B-IC-25 - Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.
Week 16:	Security Risks - Part 1-2	CSP Conceptual Framework IOC-2 - The use of computing innovations may involve risks to your personal safety and identity. IOC-2.B - Explain how computing resources can be protected and can be misused. IOC-2.B.9 - Malware is software intended to damage a computing system or to take partial control over its operation. IOC-2.C - Explain how unauthorized access to computing resources is gained. IOC-2.C.1 - Phishing is a technique that attempts to trick a user into providing personal information. That personal information can then be used to access sensitive online resources, such as bank accounts and emails. IOC-2.C.2 - Keylogging is the use of a program to record every keystroke made by a computer user in order to gain fraudulent access to passwords and other confidential information. IOC-2.C.3 - Data sent over public networks can be intercepted, analyzed, and modified. One way that this can happen is through a rogue access point. IOC-2.C.4 - A rogue access point is a wireless access point that gives unauthorized access to secure networks. IOC-2.C.5 - A malicious link can be disguised on a web page or in an email message. IOC-2.C.6 - Unsolicited emails, attachments, links, and forms in emails can be used to compromise the security of a computing system. These can come from unknown senders or from known senders whose security has been compromised. IOC-2.C.7 - Untrustworthy (often free) downloads from freeware or shareware sites can contain malware.

		IOC-2.B.8 - A computer virus is a malicious program that can copy itself and gain access to a computer in an unauthorized way. Computer viruses often attach themselves to legitimate programs and start running independently on a computer. CSTA K-12 Computer Science Standards (2017) NI - Networks & the Internet 3A-NI-05 - Give examples to illustrate how sensitive data can be affected by malware and other attacks.
	Project - Innovation Simulation - Part 3	CSP Conceptual Framework IOC-1 - While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences. IOC-1.B - Explain how a computing innovation can have an impact beyond its intended purpose. IOC-1.B.2 - Some of the ways computing innovations can be used may have a harmful impact on society, economy, or culture. IOC-1.B.3 - Responsible programmers try to consider the unintended ways their computing innovations can be used and the potential beneficial and harmful effects of these new uses. IOC-1.B.4 - It is not possible for a programmer to consider all the ways a computing innovation can be used. IOC-1.B.5 - Computing innovations have often had unintended beneficial effects by leading to advances in other fields. IOC-1.B.6 - Rapid sharing of a program or running a program with a large number of users can result in significant impacts beyond the intended purpose or control of the programmer. CSTA K-12 Computer Science Standards (2017) IC - Impacts of Computing 3A-IC-24 - Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.

		3A-IC-27 - Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields. 3B-IC-25 - Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.
	Protecting Data - Parts 1-2	CSP Conceptual Framework IOC-2 - The use of computing innovations may involve risks to your personal safety and identity. IOC-2.B - Explain how computing resources can be protected and can be misused. IOC-2.B.1 - Authentication measures protect devices and information from unauthorized access. Examples of authentication measures include strong passwords and multifactor authentication. IOC-2.B.5 - Encryption is the process of encoding data to prevent unauthorized access. Decryption is the process of decoding the data. Two common encryption approaches are: ● Symmetric key encryption involves one key for both encryption and decryption.● Public key encryption pairs a public key for encryption and a private key for decryption. The sender does not need the receiver's private key to encrypt a message, but the receiver's private key is required to decrypt the message. EXCLUSION STATEMENT(EKIOC-2.B.5): Specific mathematical procedures for encryption and decryption are beyond the scope of this course and the AP Exam. IOC-2.B.10 - All real-world systems have errors or design flaws that can be exploited to compromise them. Regular software updates help fix errors that could compromise a computing system. IOC-2.B.11 - Users can control the permissions programs have for collecting user information. Users should review the permission settings of programs to protect their privacy.

		IOC-2.B.2 - A strong password is something that is easy for a user to remember but would be difficult for someone else to guess based on knowledge of that user. IOC-2.B.3 - Multifactor authentication is a method of computer access control in which a user is only granted access after successfully presenting several separate pieces of evidence to an authentication mechanism, typically in at least two of the following categories: knowledge (something they know), possession (something they have), and inherence (something they are). IOC-2.B.4 - Multifactor authentication requires at least two steps to unlock protected information; each step adds a new layer of security that must be broken to gain unauthorized access. CSTA K-12 Computer Science Standards (2017) NI - Networks & the Internet 3A-NI-06 - Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. 3A-NI-07 - Compare various security measures, considering tradeoffs between the usability and security of a computer system. 3B-NI-04 - Compare ways software developers protect devices and information from unauthorized access.
	Project - Innovation Simulation - Part 4	CSTA K-12 Computer Science Standards (2017) IC - Impacts of Computing 3A-IC-24 - Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. 3A-IC-27 - Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields. 3B-IC-25 - Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.

	Assessment Day	
Unit 9: Create PT		
Week 17	Create PT Review the Task	
	Create PT Deep Dive	
	Create PT Make a Plan	
	Create PT-Complete the Task	
Unit 10: Algorithms		
	Algorithms Solve Problems	CSP Conceptual Framework AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.A - Express an algorithm that uses sequencing without using a programming language. AAP-2.A.1 - An algorithm is a finite set of instructions that accomplish a specific task. AAP-2.A.2 - Beyond visual and textual programming languages, algorithms can be expressed in a variety of ways, such as natural language, diagrams, and pseudocode. AAP-2.A.3 - Algorithms executed by programs are implemented using programming languages. AAP-2.A.4 - Every algorithm can be constructed using combinations of sequencing, selection, and iteration. AAP-2.B - Represent a step-by-step algorithmic process using sequential code statements. AAP-2.B.1 - Sequencing is the application of each step of an algorithm in the order in which the code statements are given. AAP-2.G - Express an algorithm that uses selection without using a programming language. AAP-2.G.1 - Selection determines which parts of an algorithm are executed based on a condition being true or false.

		AAP-2.I - For nested selection: a. Represent using nested conditional statements. b. Determine the result of nested conditional statements. AAP-2.J.1 - Iteration is a repeating portion of an algorithm. Iteration repeats a specified number of times or until a given condition is met. AAP-2.L - Compare multiple algorithms to determine if they yield the same side effect or result. AAP-2.L.1 - Algorithms can be written in different ways and still accomplish the same tasks. AAP-2.L.2 - Algorithms that appear similar can yield different side effects or results. AAP-2.L.5 - Different algorithms can be developed or used to solve the same problem.
	Algorithm Efficiency	CSP Conceptual Framework AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2.O - For algorithms involving elements of a list: a. Represent using iterative statements to traverse a list. b. Determine the result of an algorithm with list traversals. AAP-2.O.5 - Linear search or sequential search algorithms check each element of a list, in order, until the desired value is found or all elements in the list have been checked. AAP-2.P - For binary search algorithms: a. Determine the number of iterations required to find a value in a data set. b. Explain the requirements necessary to complete a binary search. AAP-2.P.1 - The binary search algorithm starts at the middle of a sorted data set of numbers and eliminates half of the data; this process repeats until the desired value is found or all elements have been eliminated. EXCLUSION STATEMENT (EK: AAP-2.P.1): Specific

		implementations of the binary search are outside the scope of the course and the AP Exam. AAP-2.P.2 - Data must be in sorted order to use the binary search algorithm. AAP-2.P.3 - Binary search is often more efficient than sequential/linear search when applied to sorted data. AAP-4 - There exist problems that the computer cannot solve, and even when the computer can solve a problem, it may not be able to do so in a reasonable amount of time. AAP-4.A - For determining the efficiency of an algorithm: a. Explain the difference between algorithms that run in reasonable time and those that do not run in reasonable time. b. Identify situations where a heuristic solution may be more appropriate. AAP-4.A.1 - A problem is a general description of a task that can (or cannot) be solved algorithmically. An instance of a problem also includes specific input. For example, sorting is a problem; sorting the list (2,3,1,7) is an instance of the problem. AAP-4.A.2 - A decision problem is a problem with a yes/no answer (e.g., is there a path from A to B?). An optimization problem is a problem with the goal of finding the "best" solution among many (e.g., what is the shortest path from A to B?). AAP-4.A.3 - Efficiency is an estimation of the amount of computational resources used by an algorithm. Efficiency is typically expressed as a function of the size of the input. EXCLUSION STATEMENT (EK AAP-4.A.3): Formal analysis of algorithms (Big-O) and formal reasoning using mathematical formulas are outside the scope of this course and the AP Exam. AAP-4.A.4 - An algorithm’s efficiency is determined through formal or mathematical reasoning. AAP-4.A.5 - An algorithm's efficiency can be informally measured by determining the number of times a statement or group of statements executes.

		AAP-4.A.6 - Different correct algorithms for the same problem can have different efficiencies. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3B-AP-10 - Use and adapt classic algorithms to solve computational problems. 3B-AP-11 - Evaluate algorithms in terms of their efficiency, correctness, and clarity.
	Unreasonable Time	CSP Conceptual Framework AAP-4 - There exist problems that the computer cannot solve, and even when the computer can solve a problem, it may not be able to do so in a reasonable amount of time. AAP-4.A - For determining the efficiency of an algorithm: a. Explain the difference between algorithms that run in reasonable time and those that do not run in reasonable time. b. Identify situations where a heuristic solution may be more appropriate. AAP-4.A.4 - An algorithm's efficiency is determined through formal or mathematical reasoning. AAP-4.A.7 - Algorithms with a polynomial efficiency or lower (constant, linear, square, cube, etc.) are said to run in a reasonable amount of time. Algorithms with exponential or factorial efficiencies are examples of algorithms that run in an unreasonable amount of time. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3B-AP-11 - Evaluate algorithms in terms of their efficiency, correctness, and clarity.
Week 18	The Limits of Algorithms	CSP Conceptual Framework AAP-4 - There exist problems that the computer cannot solve, and even when the computer can solve a problem, it may not be able to do so in a reasonable amount of time. AAP-4.A - For determining the efficiency of an algorithm: a. Explain the difference between algorithms

		that run in reasonable time and those that do not run in reasonable time. b. Identify situations where a heuristic solution may be more appropriate. AAP-4.A.2 - A decision problem is a problem with a yes/no answer (e.g., is there a path from A to B?). An optimization problem is a problem with the goal of finding the "best" solution among many (e.g., what is the shortest path from A to B?). AAP-4.A.8 - Some problems cannot be solved in a reasonable amount of time because there is no efficient algorithm for solving them. In these cases, approximate solutions are sought. AAP-4.A.9 - A heuristic is an approach to a problem that produces a solution that is not guaranteed to be optimal but may be used when techniques that are guaranteed to always find an optimal solution are impractical. EXCLUSION STATEMENT (AAP-4.A.9): Specific heuristic solutions are outside the scope of this course and the AP Exam. AAP-4.B - Explain the existence of undecidable problems in computer science. AAP-4.B.1 - A decidable problem is a decision problem for which an algorithm can be written to produce a correct output for all inputs (e.g., “Is the number even?”). AAP-4.B.2 - An undecidable problem is one for which no algorithm can be constructed that is always capable of providing a correct yes-or-no answer. EXCLUSION STATEMENT (EK AAP-4.B.2): Determining whether a given problem is undecidable is outside the scope of this course and the AP Exam. AAP-4.B.3 - An undecidable problem may have some instances that have an algorithmic solution, but there is no algorithmic solution that could solve all instances of the problem. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming

		3B-AP-11 - Evaluate algorithms in terms of their efficiency, correctness, and clarity.
	Parallel and Distributed Algorithms	CSP Conceptual Framework CSN-2 - Parallel and distributed computing leverages multiple computers to more quickly solve complex problems or process large data sets. CSN-2.A - Compare problem solutions that use sequential, parallel, and distributed computing. CSN-2.A.1 - Sequential computing is a computational model in which operations are performed in order one at a time. CSN-2.A.2 - Parallel computing is a computational model where the program is broken into multiple smaller sequential computing operations, some of which are performed simultaneously. CSN-2.A.3 - Distributed computing is a computational model in which multiple devices are used to run a program. CSN-2.A.4 - Comparing efficiency of solutions can be done by comparing the time it takes them to perform the same task. CSN-2.A.5 - A sequential solution takes as long as the sum of all of its steps. CSN-2.A.6 - A parallel computing solution takes as long as its sequential tasks plus the longest of its parallel tasks. CSN-2.A.7 - The “speedup” of a parallel solution is measured in the time it took to complete the task sequentially divided by the time it took to complete the task when done in parallel. CSN-2.B - Determine the efficiency of sequential and parallel solutions. CSN-2.B.1 - Parallel computing consists of a parallel portion and a sequential portion. CSN-2.B.2 - Solutions that use parallel computing can scale more effectively than solutions that use sequential computing.

		CSN-2.B.3 - Distributed computing allows problems to be solved that could not be solved on a single computer because of either the processing time or storage needs involved. CSN-2.B.4 - Distributed computing allows much larger problems to be solved quicker than they could be solved using a single computer. CSN-2.B.5 - When increasing the use of parallel computing in a solution, the efficiency of the solution is still limited by the sequential portion. This means that at some point, adding parallel portions will no longer meaningfully increase efficiency. CSTA K-12 Computer Science Standards (2017) AP - Algorithms & Programming 3B-AP-11 - Evaluate algorithms in terms of their efficiency, correctness, and clarity.
	Assessment Day	