Georgia Computer Science

Science II - Grades 6, 7, 8

Adopted 2021

Empowered Learner

1. **Use technology resources to increase self-direction and self-regulation in learning, including for problem solving and collaboration (e.g., using the Internet to access online resources, edit documents collaboratively).** CSS.EL.6-8.1
    1. Understand the difference between editing a shared document and suggesting edits (e.g. track changes). CSS.EL.6-8.1.1
    2. Use digital tools or platforms to organize, display, annotate, and/or share a curated collection. CSS.EL.6-8.1.2
    3. Complete an individual project (e.g., research or design) using technology resources. CSS.EL.6-8.1.3

Digital Citizen

3. **Explore computer science and computing-related careers.** CSS.DC.6-8.3
    1. Investigate a career that requires computing and technology. CSS.DC.6-8.3.1
    2. Describe how computer science enhances other career fields. CSS.DC.6-8.3.2
    3. Predict the role of computer science in future careers. CSS.DC.6-8.3.3

8. **Investigate and identify the basic components of computers and networks.** CSS.DC.6-8.8

    1. Identify the basic components of the computer by disassembling and reassembling a demonstration model personal computer (can be done 'virtually' online if demo model is not available). CSS.DC.6-8.8.1
    2. Demonstrate an understanding of key functional components (input devices, output devices, processor, operating system, software applications, memory, storage, wi-fi and/or ethernet ports, and IP addresses). CSS.DC.6-8.8.2
    3. Demonstrate an understanding of the terms and units used to describe major hardware components (RAM, ROM, GHz, MHz, GB, MB, CD, DVD, RW). CSS.DC.6-8.8.3
    4. Explain the interrelation of the operating system software, application software, and utility software, citing specific examples of each. CSS.DC.6-8.8.4
    5. Develop a basic vocabulary of networks including the Internet, wired, wireless, cellular, wi-fi, messages, packets, connections, bandwidth, broadband, firewall, hacking, cybersecurity, encryption, local area network (LAN), wide area network (WAN), and OSI model. CSS.DC.6-8.8.5
    6. Demonstrate an understanding of the fundamental concepts for how computers process programming commands (hex, binary language, sequence of commands, conditional structures, looping structures). CSS.DC.6-8.8.6

Networks and the Internet

9. Investigate ways to differentiate networks and how they are used in business and industry. CSS.DC.6-8.9
    1. Create diagrams to illustrate types of network topologies to include star, ring, bus, mesh, and hybrid. CSS.DC.6-8.9.1
    2. Differentiate networks based on coverage area including local area network (LAN), wide area network (WAN), and personal area network (PAN) CSS.DC.6-8.9.2
    3. Differentiate between different network mediums including Wi-fi, wired, satellite, and microwave. CSS.DC.6-8.9.3
10. Evaluate and provide a rationale for the levels of the Open Systems Interconnection (OSI) model. CSS.DC.6-8.10
    1. Summarize from multiple sources the physical and digital aspects of computing networks. CSS.DC.6-8.10.1
    2. Trace the layers required to transmit data from one node to another (the OSI model). CSS.DC.6-8.10.2
    3. Construct and explain the basic functions of the OSI model. CSS.DC.6-8.10.3
11. Examine the basics of cybersecurity needs for business, government, and organizations. CSS.DC.6-8.11
    1. List and define the elements of the confidentiality, integrity, and availability (CIA) triad. CSS.DC.6-8.11.1
    2. Explain components of access control: Identification, Authentication, Authorization, Accountability, and Non-repudiation. CSS.DC.6-8.11.2
    3. Identify the characteristics of strong vs. weak passwords in data and identity security. CSS.DC.6-8.11.3
    4. List and describe the basic steps in security risk management. CSS.DC.6-8.11.4
    5. Develop a logical argument for the importance of physical security. CSS.DC.6-8.11.5
12. Cite evidence regarding the principles of cybersecurity and basic mechanisms used for protecting data and resources. CSS.DC.6-8.12
    1. Define the cybersecurity first principles of least privilege, minimization, abstraction, domain separation, process isolation, information hiding, layering, simplicity, modularity, and resource encapsulation. CSS.DC.6-8.12.1
    2. Apply concepts related to the principles behind encryption, including the purpose of cryptography, hashing, and steganography. CSS.DC.6-8.12.2
    3. Draw conclusions illustrating a basic understanding of internet protocol (IP) packets, ports and network transmission. CSS.DC.6-8.12.3
    4. Summarize from multiple sources a basic understanding of anti-malware, firewalls, intrusion detection system/intrusion prevention system (IDS/IPS), and virtual private network (VPN). CSS.DC.6-8.12.4

13. Analyze and describe the characteristics of cybersecurity ethics, digital citizenship, and laws governing privacy. CSS.DC.6-8.13

    1. Explain the differences between a white hat (ethical) hacker and a black hat (unethical) hacker. CSS.DC.6-8.13.1
    2. Cite evidence regarding the practice of ethical digital decision-making, including plagiarism, copyright law, and software licensing types (freeware, public domain, shareware, etc.). CSS.DC.6-8.13.2
    3. Summarize and provide examples regarding security and privacy laws and their impact on society, citing recent cases. CSS.DC.6-8.13.3
    4. Analyze cyberbullying to include legal and social consequences CSS.DC.6-8.13.4
    5. Develop a set of guidelines to prevent cyberbullying. CSS.DC.6-8.13.5
    6. Develop arguments for policy-driven and technology-driven security. CSS.DC.6-8.13.6

Innovative Designer and Creator

21. **Develop a plan to create, design, and build a website with digital content to a specific target market.** CSS.KC.6-8.21

    1. Identify the objectives (e.g., increase sales, promote new products, increase company awareness, target new customers) for the website's target market. CSS.KC.6-8.21.1
    2. Specify website requirements, including timeline and resources, and organize them into a requirements document. CSS.KC.6-8.21.2
    3. Find and evaluate similar websites (in terms of overall function and layout) using an evaluation instrument for side-by-side comparison. Consider major design elements (ease of use, responsiveness, adaptability to mobile, tablet and desktop, etc.). CSS.KC.6-8.21.3
    4. Evaluate a variety of web design tools and development platforms using an evaluation instrument and choose the appropriate platform. CSS.KC.6-8.21.4

22. **Design digital products that reveal a professional layout and look by applying design principles to produce professional quality digital products.** CSS.KC.6-8.22

    1. Identify graphical elements and the appropriate use of elements on a web site. CSS.KC.6-8.22.1
    2. Explore and apply color principles to digital products. CSS.KC.6-8.22.2
    3. Establish a brand through consistent use of graphics, color, layout and text. CSS.KC.6-8.22.3
    4. Analyze the look and layout of a website based on the first impression of content and page elements. Get feedback from independent people and incorporate where appropriate. CSS.KC.6-8.22.4

23. **Create a single functional web page using a web development platform based on a design mockup and user requirements.** CSS.KC.6-8.23

    1. Create and edit images and graphics for website publication. CSS.KC.6-8.23.1
    2. Plan, produce, and edit digital audio for website publication. CSS.KC.6-8.23.2
    3. Plan, produce, edit, and post a multimedia-rich video project to a website. CSS.KC.6-8.23.3
    4. Plan, produce, and edit animations for website publication. CSS.KC.6-8.23.4

24. **Develop and use a test plan to debug each new website version to ensure it runs as intended and meets the end-user requirements for a responsive site.** CSS.KC.6-8.24

    1. Create a test and debug plan. Resolve issues and fix any errors that surface during the test and debug process. CSS.KC.6-8.24.1
    2. Create an end user testing plan, get user feedback, and incorporate feedback into the final website. CSS.KC.6-8.24.2
    3. Prepare website for publishing and promotion. CSS.KC.6-8.24.3

25. **Develop a plan to create, design, and build a game with digital content for a specific target market.** CSS.KC.6-8.25

    1. Explore various game types including role-playing games (RPG), real-time strategy (RTS), simulations, puzzles, educational, massively multiplayer online (MMO), and others. CSS.KC.6-8.25.1
    2. Create a Game Design Document (GDD), which includes, characters, story, theme, and gameplay mechanics. CSS.KC.6-8.25.2

26. **Develop a visual model of a game from the Game Design Document (GDD).** CSS.KC.6-8.26

    1. Create storyboards from the GDD that demonstrate game progression and consistent use of a theme. CSS.KC.6-8.26.1
    2. Use the GDD to design the wireframes and comprehensive layout for the user experience (UX). CSS.KC.6-8.26.2

27. **Create a functional game, using a game development platform, based on the storyboards, wireframes, and comprehensive layout.** CSS.KC.6-8.27

    1. Create game elements, backgrounds, and characters. CSS.KC.6-8.27.1
    2. Use scripting languages to create desired game mechanics, and to control the environment, user interface (UI), and character behaviors. CSS.KC.6-8.27.2
    3. Plan, produce, and edit graphics and animations for game publication. CSS.KC.6-8.27.3
    4. Plan, produce, and edit digital audio for game publication. CSS.KC.6-8.27.4

28. **Develop and use a test plan to debug use each time a version of the game is released to ensure it runs as intended and meets the end-user requirements.** CSS.KC.6-8.28

    1. Create a test and debug plan. Resolve any issues and fix any errors that surface during the test and debug process. CSS.KC.6-8.28.1
    2. Create an end user testing plan, get user feedback, and incorporate feedback into the final game. CSS.KC.6-8.28.2
    3. Prepare final game for publishing prior to publishing to the target audience. CSS.KC.6-8.28.3

Computational Thinker

Recognizing and Defining Computational Problems

32. Develop through application, logical observations relative to computational thinking procedures to analyze and solve problems current to everyday life. CSS.CT.6-8.32

    1. Identify characteristics of computational thinking (decomposition, pattern recognition, algorithmic thinking and abstraction). CSS.CT.6-8.32.1
    2. Explain issues and analyze routine hardware and software problems current to everyday life. CSS.CT.6-8.32.2
    3. Apply troubleshooting concepts to issues regarding compatibility, data, and identity. CSS.CT.6-8.32.3
    4. Describe ways to resolve operational problems caused by hardware errors. CSS.CT.6-8.32.4
    5. Explain how technology can create ethical and legal issues in the business world and a technology-based society and how it can be used to solve & manage those issues. CSS.CT.6-8.32.5

33. Utilize computational thinking to solve problems. CSS.CT.6-8.33

    1. Make observations and organize the concepts of modularity, including functions and methods, as it relates to programming code reusability and cloud computing in the software industry. CSS.CT.6-8.33.1
    2. Develop a working vocabulary of computational thinking including sequences, algorithms, binary, pattern matching, decomposition, abstraction, parallelization, data, automation, data collection, data analysis, boolean, integer, branches (if...then...else), and iteration {loops (For, While)}. CSS.CT.6-8.33.2
    3. Analyze the problem-solving process, the input-process-output-storage model of a computer, and how computers help humans solve problems. CSS.CT.6-8.33.3
    4. Develop an algorithm to decompose a problem of a daily task. CSS.CT.6-8.33.4

34. Recognize when to use the same solution for multiple problems. CSS.CT.6-8.34

Data and Information

35. Evaluate the storage and representation of data; Analyze how data is collected with both computational and non-computational tools and processes. CSS.CT.6-8.35
    1. Discuss binary numbers, logic, sets, and functions and their application to computer science. CSS.CT.6-8.35.1
    2. Explain that searches may be enhanced by using Boolean logic (e.g., using "not", "or", "and"). CSS.CT.6-8.35.2

Algorithms

36. Understand and use the basic steps in algorithmic problem solving in computing and other authentic applications. CSS.CT.6-8.36
    1. Select basic steps to solve algorithmic problems. CSS.CT.6-8.36.1
    2. Evaluate basic steps of algorithmic problem solving to design solutions. CSS.CT.6-8.36.2
    3. Solve algorithmic problems of increasing complexity. CSS.CT.6-8.36.3

Programming

37. Use and compare simple coding control structures (e.g., if-then, loops) CSS.CT.6-8.37
    1. Use a visual block-based and/or text-based programming language individually and collaboratively to solve problems of increasing complexity. CSS.CT.6-8.37.1
    2. Create a program individually and collaboratively using a text-based programming. language; Identify variables and compare the types of data stored as variables. CSS.CT.6-8.37.2

Creating Computational Artifacts

38. Consider the purpose of computational artifacts for practical use, personal expression, and/or societal impact. CSS.CT.6-8.38
    1. Compare and contrast examples of high level and low-level programming languages. CSS.CT.6-8.38.1
    2. Investigate the notion of hierarchy in computing including high level languages, translations, instruction sets, and logic circuits. CSS.CT.6-8.38.2
    3. Develop problem solutions using a programming language, including all of the following: looping behavior, conditional statements, expressions, variables, and functions. CSS.CT.6-8.38.3

Testing and Refining Computational Artifacts

39. Test computational artifacts systematically by considering multiple scenarios and using test cases. CSS.CT.6-8.39

Human Computer Interaction

40. Describe how humans and machines interact to accomplish tasks that cannot be accomplished by either alone. CSS.CT.6-8.40
    1. Identify what distinguishes humans from machines focusing on human intelligence versus machine intelligence (e.g., robot motion, speech and language understanding, and computer vision); Explain why some tasks can be accomplished more easily by computers. CSS.CT.6-8.40.1
    2. Describe ways in which computers use models of intelligent behavior (e.g., robot motion, speech and language understanding, and computer vision) and how they differ (e.g., emotional decision making versus logical decisions, common sense, literal versus abstract). CSS.CT.6-8.40.2
    3. Design and demonstrate the use of a device (e.g., robot, e-textile) to accomplish a task, individually and collaboratively. CSS.CT.6-8.40.3

Creative Communicator

Collaborating Around Computing

41. Use online resources to participate in collaborative activities for the purpose of developing solutions or products. CSS.CT.6-8.41
42. Improve teamwork and collaboration skills: providing useful feedback, integrating feedback, understanding, and accepting multiple perspectives. CSS.CT.6-8.42
    1. Understand the difference between CC and BCC as well as Reply and Reply All and when to use each appropriately. CSS.CT.6-8.42.1
43. Collaborate productively and recognize the value of working with individuals of varying perspectives, skills, and backgrounds. CSS.CT.6-8.43
    1. Set and implement equitable expectations and workloads when working in teams. CSS.CT.6-8.43.1
44. Demonstrate correct keyboarding techniques while increasing speed and maintaining accuracy. CSS.CT.6-8.44
45. Use productivity technology tools (e.g. word processing, spreadsheet, presentation software) for individual and collaborative writing, communication, and publishing activities. CSS.CT.6-8.45

Global Collaborator

Fostering an Inclusive Computing Culture

46. Recognize that equitable access to computing benefits society as a whole. CSS.CT.6-8.46
47. Consider others' perspectives as well as one's own perspective when developing computational solutions. CSS.CT.6-8.47
48. Consider the needs of a variety of end users regarding accessibility and usability. CSS.CT.6-8.48
49. Use software applications to collaborate and create authentic products. CSS.CT.6-8.49
    1. Identify and utilize the appropriate software application for productivity. CSS.CT.6-8.49.1
    2. Use various applications in a professional manner to share and communicate with peers and teachers. CSS.CT.6-8.49.2
    3. Share documents created using word processing, presentation, and spreadsheet software. CSS.CT.6-8.49.3
    4. Create original works using software applications in a collaborative manner. CSS.CT.6-8.49.4
    5. Collaborate in small groups to create and edit online documents in real time. CSS.CT.6-8.49.5
    6. Identify and use appropriate file sharing strategies (e.g., copy and paste, links, posts, and attachments). CSS.CT.6-8.49.6