Georgia Computer Science

Science I - 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

2. **Understand benefits and risks of digital citizenship and practices safe, responsible, legal, and ethical behavior while using technology tools and resources, especially related to personal information.** CSS.DC.6-8.2
    1. Understand that digital content is permanent and cannot be deleted. CSS.DC.6-8.2.1
    2. Discuss the benefits and risks of using password management systems and storing personal information online. CSS.DC.6-8.2.2
    3. Understand how browser settings such as cookies collect data and track personal information. CSS.DC.6-8.2.3

4. **Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution.** CSS.DC.6-8.4

5. **Understand the pervasiveness and tradeoffs of computers and computing in daily life.** CSS.DC.6-8.5
    1. Find and adjust privacy settings for online accounts; discuss the positive and negative effects that social media can have on one's life. CSS.DC.6-8.5.1
    2. Recognize how the overuse of technology can impact mental, physical, and emotional health. CSS.DC.6-8.5.2
    3. Identify phishing emails, insecure websites (e.g., not https), and risky links when on the Internet; build a positive digital footprint (e.g., blog, website, social media). CSS.DC.6-8.5.3

6. **Apply strategies for troubleshooting hardware and software problems that may occur during use.** CSS.DC.6-8.6
    1. Choose the correct settings for printing and troubleshoot common printing issues (printer not turned on, printer not connected). CSS.DC.6-8.6.1
    2. Connect devices (including internet/WIFI and Bluetooth) and troubleshoot common connection issues (e.g., disrupted Internet connection, website that is not loading). CSS.DC.6-8.6.2
    3. Manage and deploy software updates and troubleshoot simple software issues (i.e., look up solutions to issues). CSS.DC.6-8.6.3

7. **Explore the relationship between computer hardware and software.** CSS.DC.6-8.7
    1. Develop a working vocabulary of embedded computing including digital, analog, events, microcontrollers, sensors, LED, switch, servo, cloud computing, and internet of things. CSS.DC.6-8.7.1
    2. Investigate how software interacts with hardware in the boot process. CSS.DC.6-8.7.2
    3. Analyze and explain how computers communicate information with simple hardware inputs and outputs. CSS.DC.6-8.7.3
    4. Create a product that analyzes how simple computer hardware can be used to develop innovative new products that interact with the physical world. CSS.DC.6-8.7.4
    5. Design a computer program that senses something in the real world and changes an output based on the input. CSS.DC.6-8.7.5

Knowledge Constructor

Data and Analysis

14. Evaluate the accuracy, relevance, appropriateness, comprehensiveness, and biases that occur in electronic information sources. CSS.KC.6-8.14
    1. Demonstrate an understanding of the credibility, bias, accuracy, relevance, age appropriateness, and comprehensiveness of electronic information sources. CSS.KC.6-8.14.1
    2. Evaluate and discuss of the credibility, bias, accuracy, relevance, age appropriateness, and comprehensiveness of electronic information sources. CSS.KC.6-8.14.2
    3. Apply strategies for determining the reliability of information found on the Internet. CSS.KC.6-8.14.3
15. Gather, manipulate, and analyze data using a variety of digital tools to identify solutions and make informed decisions. CSS.KC.6-8.15
    1. Gather data and calculate numerical equations using spreadsheet formulas and functions. CSS.KC.6-8.15.1
    2. Use spreadsheet data to create tables, charts, and graphs. CSS.KC.6-8.15.2
    3. Use spreadsheets and databases to make predictions, solve problems, and draw conclusions. CSS.KC.6-8.15.3
16. Traverse online environments using critical thinking to find valid sources of information. CSS.KC.6-8.16
17. Analyze various ways to visually represent data. CSS.KC.6-8.17
    1. Interpret tables, charts, and graphs created by someone else. CSS.KC.6-8.17.1
    2. Discuss design decisions in choosing between text, tables, charts, and graphs. CSS.KC.6-8.17.2
    3. Discuss design decisions to make visualizations of data clear and concise. CSS.KC.6-8.17.3

Innovative Designer and Creator

18. **Recognize that there may be multiple approaches to solving a problem.** CSS.KC.6-8.18

19. **Approach problem solving iteratively, using a cyclical process.** CSS.KC.6-8.19

20. **Design, develop, debug and implement computer programs.** CSS.KC.6-8.20

    1. Develop a working vocabulary of programming including flowcharting and/or storyboarding, coding, debugging, user interfaces, usability, variables, lists, loops, conditionals, programming language, events. CSS.KC.6-8.20.1
    2. Utilize the design process to brainstorm, implement, test, and revise an idea. CSS.KC.6-8.20.2
    3. Cite evidence on how computers represent data and media (sounds, images, video, etc.). CSS.KC.6-8.20.3
    4. Design a user interface and test with other users using a paper prototype. CSS.KC.6-8.20.4
    5. Implement a simple algorithm in a computer program. CSS.KC.6-8.20.5
    6. Develop an event driven program. CSS.KC.6-8.20.6
    7. Create a program that accepts user and/or sensor input and stores the result in a variable. CSS.KC.6-8.20.7
    8. Create a computer program that implements a loop. CSS.KC.6-8.20.8
    9. Develop a program that makes a decision based on data or user input. CSS.KC.6-8.20.9
    10. Debug a program with an error. CSS.KC.6-8.20.10

29. **Create digital artifacts to address a current issue requiring resolution.** CSS.KC.6-8.29

    1. Summarize ethical, privacy, and legal issues of a digital world using current case studies. CSS.KC.6-8.29.1
    2. Collaborate as a team to develop an artifact that represents multiple perspectives regarding a global crisis. CSS.KC.6-8.29.2
    3. Analyze and explain the functionality and suitability (or appropriateness) of a computational artifact. CSS.KC.6-8.29.3
    4. Develop a program for creative expression or to satisfy personal curiosity which may have visual, audible, and/or tactile results. CSS.KC.6-8.29.4
    5. Develop a program specifically with the goal of solving a problem, creating new knowledge, or helping people, organizations, or society. CSS.KC.6-8.29.5

Computational Thinker

Recognzing and Defining Computational Problems

30. Identify sub-problems to consider while addressing a larger problem. CSS.CT.6-8.30
31. Recognize when it is appropriate to solve a problem computationally; Make sense of computational problems and persevere in solving them. CSS.CT.6-8.31
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