

3–5 3-5.CS.1 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.CS.1 Describe how computing devices connect to other components to form a system. Computing Systems Devices Communicating 7.2 Descriptive Statement Computing devices often depend on other devices or components. Students describe physical and wireless connections to other components, including both input devices (e.g., keyboards, sensors, remote controls, microphones) and output devices (e.g., 3D printers, monitors, speakers). For example, students could describe the relationship among the heart, lungs, muscles, blood, and oxygen during physical activity and then compare this to how a mouse, keyboard, printer, and desktop computer connect and interact to allow for input, processing, and output (P.E.3.4.7). Alternatively, when describing how light reflected from objects enters the eye and is then transferred to the brain to construct a visual image, students could compare this to a computing system that uses programming to construct a visual image when data is transferred and constructed/reconstructed through a keyboard, camera, or other components (CA NGSS: 4-PS4-2). California Department of Education California K–12 Computer Science Standards | 51

3-5.CS.2 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.CS.2 Demonstrate how computer hardware and software work together as a system to accomplish tasks. Computing Systems Hardware & Software Abstraction 4.4 Descriptive Statement Hardware and software are both needed to accomplish tasks with a computing device. Students create a model to illustrate ways in which hardware and software work as a system. Students could draw a model on paper or in a drawing program, program an animation to demonstrate it, or demonstrate it by acting this out in some way. At this level, a model should only include the basic elements of a computer system, such as input, output, processor, sensors, and storage. For example, students could create a diagram or flow chart to indicate how a keyboard, desktop computer, monitor, and word processing software interact with each other. The keyboard (hardware) detects a key press, which the operating system and word processing application (software) displays as a new character that has been inserted into the document and is visible through the monitor (hardware). Students could also create a model by acting out the interactions of these different hardware and software components. Alternatively, when describing that animals and people receive different types of information through their senses, process the information in their brain, and respond to the information in different ways, students could compare this to the interaction of how the information traveling through a computer from mouse to processor are similar to signals sent through the nervous system telling our brain about the world around us to prompt responses (CA NGSS: 4-LS1-2). 52 | California K–12 Computer Science Standards California Department of Education

3-5.CS.3 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.CS.3 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. Computing Systems Troubleshooting Testing 6.2 Descriptive Statement Although computing systems vary, common troubleshooting strategies can be used across many different systems. Students use troubleshooting strategies to identify problems that could include a device not responding, lacking power, lacking a network connection, an app crashing, not playing sounds, or password entry not working. Students use and develop various solutions to address these problems. Solutions may include rebooting the device, checking for power, checking network

availability, opening and closing an app, making sure speakers are turned on or headphones are plugged in, and making sure that the caps lock key is not on. For example, students could prepare for and participate in a collaborative discussion in which they identify and list computing system problems and then describe common successful fixes (CA CCSS for ELA/Literacy SL.3.1, SL.4.1, SL.5.1). Alternatively, students could write informative/explanatory texts, create a poster, or use another medium of communication to examine common troubleshooting strategies and convey these ideas and information clearly (CA CCSS for ELA/Literacy W.3.2, W.4.2, W.5.2). California Department of Education California K–12 Computer Science Standards | 53

3-5.NI.4 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.NI.4 Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the internet, and reassembled at the destination. Networks & the Internet Network Communication & Organization Abstraction 4.4 Descriptive Statement Information is sent and received over physical or wireless paths. It is broken down into smaller pieces called packets, which are sent independently and reassembled at the destination. Students demonstrate their understanding of this flow of information by, for instance, drawing a model of the way packets are transmitted, programming an animation to show how packets are transmitted, or demonstrating this through an unplugged activity in which they physically act this out. For example, students could design a structure using building blocks or other materials with the intention of re-engineering it in another location, just as early Americans did after the intercontinental railroad was constructed in the 1850s (HSS.4.4.1, 4.4.2). Students could deconstruct the designed structure, place materials into specific containers (or plastic bags/brown paper bags/etc.), and develop instructions on how to recreate the structure once each container arrives at its intended destination (CA NGSS: 3-5-ETS1). For example, students could cut up a map of the United States by state lines. Students could then place the states in envelopes and transmit the “packets” through a physical network, represented by multiple students spreading out in arms reach of at least two others. At the destination, the student who receives the packets reassembles the individual states back into a map of the United States (HSS 5.9). Alternatively, students could perform a similar activity with a diatonic scale, cutting the scale into individual notes. Each note, in order, should be placed into a numbered envelope based on its location on the scale. These envelopes can be transmitted across the network of students and reassembled at the destination (VAPA Music 4.1.2). 54 | California K–12 Computer Science Standards California Department of Education

3-5.NI.5 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.NI.5 Describe physical and digital security measures for protecting personal information. Networks & the Internet Cybersecurity Computational Problems 3.1 Descriptive Statement Personal information can be protected physically and digitally. Cybersecurity is the protection from unauthorized use of electronic data, or the measures taken to achieve this. Students identify what personal information is and the reasons for protecting it. Students describe physical and digital approaches for protecting personal information such as using strong passwords and biometric scanners. For example, students could engage in a collaborative discussion orally or in writing regarding topics that relate to personal cybersecurity issues. Discussion topics could be based on current events related to cybersecurity or topics that are applicable to students, such as the necessity of backing up data to guard against loss, how to create strong passwords and the importance of not sharing passwords, or why we should keep

operating systems updated and use anti-virus software to protect data and systems. Students could also discuss physical measures that can be used to protect data including biometric scanners, locked doors, and physical backups (CA CCSS for ELA/Literacy SL.3.1, SL.4.1, SL.5.1). California Department of Education California K–12 Computer Science Standards | 55

3-5.NI.6 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.NI.6 Create patterns to protect information from unauthorized access. Networks & the Internet Cybersecurity Abstraction 4.4 Descriptive Statement Encryption is the process of converting information or data into a code, especially to prevent unauthorized access. At this level, students use patterns as a code for encryption, to protect information. Patterns should be decodable to the party for whom the message is intended, but difficult or impossible for those with unauthorized access. For example, students could create encrypted messages via flashing a flashlight in Morse code. Other students could decode this established language even if it wasn't meant for them. To model the idea of protecting data, students should create their own variations on or changes to Morse code. This ensures that when a member of that group flashes a message only other members of their group can decode it, even if other students in the room can see it (CA NGSS: 4-PS4-3). Alternatively, students could engage in a CS Unplugged activity that models public key encryption: One student puts a paper containing a written secret in a box, locks it with a padlock, and hands the box to a second student. Student 2 puts on a second padlock and hands it back. Student 1 removes her lock and hands the box to student 2 again. Student 2 removes his lock, opens the box, and has access to the secret that student 1 sent him. Because the box always contained at least one lock while in transit, an outside party never had the opportunity to see the message and it is protected. 56 | California K–12 Computer Science Standards California Department of Education

3-5.DA.7 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.DA.7 Explain that the amount of space required to store data differs based on the type of data and/or level of detail. Data & Analysis Storage Abstraction 4.2 Descriptive Statement All saved data requires space to store it, whether locally or not (e.g., on the cloud). Music, images, video, and text require different amounts of storage. Video will often require more storage and different format than music or images alone because video combines both. The level of detail represented by that data also affects storage requirements. For instance, two pictures of the same object can require different amounts of storage based upon their resolution, and a high-resolution photo could require more storage than a low-resolution video. Students select appropriate storage for their data. For example, students could create an image using a standard drawing app. They could save the image in different formats (e.g., .png, .jpg, .pdf) and compare file sizes. They should also notice that different file sizes can result in differences in quality or resolution (e.g., some pictures could be more pixelated while some could be sharper). Alternatively, in an unplugged activity, students could represent images by coloring in squares within a large grid. They could model how a larger grid requires more storage but also represents a clearer image (i.e., higher resolution). California Department of Education California K–12 Computer Science Standards | 57

3-5.DA.8 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.DA.8

Organize and present collected data visually to highlight relationships and support a claim. Data & Analysis Collection Visualization & Transformation Communicating 7.1 Descriptive Statement Raw data has little meaning on its own. Data is often sorted or grouped to provide additional clarity. Organizing data can make interpreting and communicating it to others easier. Data points can be clustered by a number of commonalities. The same data could be manipulated in different ways to emphasize particular aspects or parts of the data set. For example, students could create and administer electronic surveys to their classmates. Possible topics could include favorite books, family heritage, and after school activities. Students could then create digital displays of the data they have collected such as column histogram charts showing the percent of respondents in each grade who selected a particular favorite book. Finally, students could make quantitative statements supported by the data such as which books are more appealing to specific ages of students. As an extension, students could write an opinion piece stating a claim and supporting it with evidence from the data they collected (CA CCSS for Mathematics 3.MD.3, 4.MD.4, 5.MD.2) (CA CCSS for ELA/Literacy W.3.1, W.4.1, W.5.1). Alternatively, students could represent data in tables and graphical displays to describe weather experienced in the last several years. They could select the type of graphical display based on the specific data represented (e.g., daily high/low temperatures on a scatter plot, average temperatures for a month across years in a column chart). Students could then make a claim about expected weather in future months based on the data (CA NGSS: 3-ESS2-1). 58 | California K–12 Computer Science Standards California Department of Education

3-5.DA.9 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.DA.9 Use data to highlight and/or propose relationships, predict outcomes, or communicate ideas. Data & Analysis Inference & Models Communicating 7.1 Descriptive Statement The accuracy of data analysis is related to how the data is represented. Inferences or predictions based on data are less likely to be accurate if the data is insufficient, incomplete, or inaccurate or if the data is incorrect in some way. Additionally, people select aspects and subsets of data to be transformed, organized, and categorized. Students should be able to refer to data when communicating an idea, in order to highlight and/or propose relationships, predict outcomes, highlight different views and/or communicate insights and ideas. For example, students can be provided a scenario in which they are city managers who have a specific amount of funds to improve a city in California. Students can collect data of a city concerning land use, vegetation, wildlife, climate, population density, services and transportation (HSS.4.1.5) to determine and present what area needs to be focused on to improve a problem. Students can compare their data and planned use of funds with peers, clearly communicating or predict outcomes based on data collected (CA CCCS for ELA/Literacy SL.3.1, SL.4.1, SL.5.1). Alternatively, students could record the temperature at noon each day to show that temperatures are higher in certain months of the year. If temperatures are not recorded on non-school days or are recorded incorrectly, the data would be incomplete and ideas being communicated could be inaccurate. Students may also record the day of the week on which the data was collected, but this would have no relevance to whether temperatures are higher or lower. In order to have sufficient and accurate data on which to communicate the idea, students might use data provided by a governmental weather agency (CA NGSS: 3-ESS2-1). California Department of Education California K–12 Computer Science Standards | 59

3-5.AP.10 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.AP.10

Compare and refine multiple algorithms for the same task and determine which is the most appropriate. Algorithms & Programming Algorithms Testing, Computational Problems 6.3, 3.3 Descriptive Statement Different algorithms can achieve the same result, though sometimes one algorithm might be more appropriate for a specific solution. Students examine different ways to solve the same task and decide which would be the better solution for the specific scenario. For example, students could use a map and create multiple algorithms to model the early land and sea routes to and from European settlements in California. They could then compare and refine their algorithms to reflect faster travel times, shorter distances, or avoid specific characteristics, such as mountains, deserts, ocean currents, and wind patterns (HSS.4.2.2). Alternatively, students could identify multiple algorithms for decomposing a fraction into a sum of fractions with the same denominator and record each decomposition with an equation (e.g., $2\frac{1}{8} = 1 + 1 + \frac{1}{8} = \frac{8}{8} + \frac{8}{8} + \frac{1}{8}$). Students could then select the most efficient algorithm (e.g., fewest number of steps) (CA CCSS for Mathematics 4.NF.3b). Additionally, students could compare algorithms that describe how to get ready for school and modify them for supporting different goals including having time to care for a pet, being able to talk with a friend before classes start, or taking a longer route to school to accompany a younger sibling to their school first. Students could then write an opinion piece, justifying with reasons their selected algorithm is most appropriate (CA CCSS for ELA/Literacy W.3.1, W.4.1, W.5.1). 60 | California K–12 Computer Science Standards California Department of Education

3-5.AP.11 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.AP.11 Create programs that use variables to store and modify data. Algorithms & Programming Variables Creating 5.2 Descriptive Statement Variables are used to store and modify data. Students use variables in programs they create. At this level, students may need guidance in identifying when to create variables (i.e., performing the abstraction). For example, students could create a game to represent predators and prey in an ecosystem. They could declare a “score” variable, assign it to 0 at the start of the game, and add 1 (increment) the score each time the predator captures its prey. They could also declare a second “numberOfLives” variable, assign it to 3 at the start of the game, and subtract 1 (decrement) each time a prey is captured. They could program the game to end when “numberOfLives” equals 0 (CA NGSS: 5-LS2-1) (CA CCSS for Mathematics 5.OA.3). Alternatively, when students create programs to draw regular polygons, they could use variables to store the line size, line color, and/or side length. Students can extend learning by creatively combining a variety of polygons to create digital artwork, comparing and contrasting this to another work of art made by the use of different art tools and media, such as watercolor or tempera paints (CA CCSS for Mathematics 3.G.1) (VAPA Visual Arts 3.1.4). California Department of Education California K–12 Computer Science Standards | 61 3-5.AP.12 Grade Standard Identifier Standard Framework Alignment: Concept Framework

3-5.AP.12 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.AP.12 Create programs that include events, loops, and conditionals. Algorithms & Programming Control Creating 5.2 Descriptive Statement Control structures specify the order (sequence) in which instructions are executed within a program and can be combined to support the creation of more complex programs. Events allow portions of a program to run based on a specific action. Conditionals allow for the execution of a portion of code in a program when a certain condition is true. Loops allow for the

repetition of a sequence of code multiple times. For example, students could program an interactive map of the United States of America. They could use events to initiate a question when the user clicks on a state and conditionals to check whether the user input is correct. They could use loops to repeat the question until the user answers correctly or to control the length of a “congratulations” scenario that plays after a correct answer (HSS.5.9). Alternatively, students could write a math fluency game that asks products of two one-digit numbers and then uses a conditional to check whether or not the answer that was entered is correct. They could use a loop to repeatedly ask another question. They could use events to allow the user to click on a green button to play again or a red button to end the game (CA CCSS for Mathematics 3.OA.7). Additionally, students could create a program as a role-playing game based on a literary work. Loops could be used to animate a character’s movement. When reaching a decision point in the story, an event could initiate the user to type a response. A conditional could change the setting or have the story play out differently based on the user input (CA CCSS for ELA/Literacy RL.5.3). 62 | California K–12 Computer Science Standards California Department of Education

3-5.AP.13 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.AP.13 Decompose problems into smaller, manageable tasks which may themselves be decomposed. Algorithms & Programming Modularity Computational Problems 3.2 Descriptive Statement Decomposition is the act of breaking down tasks into simpler tasks. This manages complexity in the problem solving and program development process. For example, students could create an animation to represent a story they have written. Students write a story and then break it down into different scenes. For each scene, they would select a background, place characters, and program actions in that scene (CA CCSS for ELA/ Literacy W.3.3, W.4.3, W.5.3). Alternatively, students could create a program to allow classmates to present data collected in an experiment. For example, if students collected rain gauge data once per week for 3 months, students could break down the program tasks: (1) ask the user to input 12 weeks’ worth of data, (2) process the data (e.g., add the first four entries to calculate the rain amount for month 1, convert to metric system measurements), and (3) direct the creation or resizing of objects (e.g., one rectangular chart bar for each month) to represent the total number of rainfall for that month (CA NGSS: 3-ETS-1-2) (CA CCSS for Mathematics 3.MD.2). California Department of Education California K–12 Computer Science Standards | 63

3-5.AP.14 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.AP.14 Create programs by incorporating smaller portions of existing programs, to develop something new or add more advanced features. Algorithms & Programming Modularity, Program Development Abstraction, Creating 4.2, 5.3 Descriptive Statement Programs can be broken down into smaller parts, which can be incorporated into new or existing programs. Students incorporate predefined functions into their original designs. At this level, students do not need to understand all of the underlying implementation details of the abstractions that they use. For example, students could use code from a ping pong animation to make a ball bounce in a new basketball game. They could also incorporate code from a single-player basketball game to create a two-player game with slightly different rules. Alternatively, students could remix an animated story and add their own conclusion and/or additional dialogue (CA CCSS for ELA/Literacy W.3.3.B, W.3.3.D, W.4.3.B, W.4.3.E, W.5.3.B, W.5.3.E). Additionally, when creating a game that occurs on the moon or planets, students could incorporate and modify code

that simulates gravity on Earth. They could modify the strength of the gravitational force based on the mass of the planet or moon (CA NGSS: 5-PS2-1). 64 | California K–12 Computer Science Standards
California Department of Education

3-5.AP.15 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.AP.15 Use an iterative process to plan and develop a program by considering the perspectives and preferences of others. Algorithms & Programming Program Development Inclusion, Creating 1.1, 5.1 Descriptive Statement Planning is an important part of the iterative process of program development. Students gain a basic understanding of the importance and process of planning before beginning to write code for a program. They plan the development of a program by outlining key features, time and resource constraints, and user expectations. Students should document the plan as, for example, a storyboard, flowchart, pseudocode, or story map. For example, students could collaborate with a partner to plan and develop a program that graphs a function. They could iteratively modify the program based on feedback from diverse users, such as students who are color blind and may have trouble differentiating lines on a graph based on the color (CA CCSS for Mathematics 5.G.1, 5.G.2). Alternatively, students could plan as a team to develop a program to display experimental data. They could implement the program in stages, generating basic displays first and then soliciting feedback from others on how easy it is to interpret (e.g., are labels clear and readable?, are lines thick enough?, are titles understandable?). Students could iteratively improve their display to make it more readable and to better support the communication of the finding of the experiment (NGSS.3-5-ETS1-1, 3-5-ETS1-2, 3-5-ETS1-3). California Department of Education California K–12 Computer Science Standards | 65

3-5.AP.16 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.AP.16 Observe intellectual property rights and give appropriate attribution when creating, remixing, or combining programs. Algorithms & Programming Program Development Creating, Communicating 5.2, 7.3 Descriptive Statement Intellectual property rights can vary by country, but copyright laws give the creator of a work a set of rights and prevents others from copying the work and using it in ways that they may not like. Students consider common licenses that place limitations or restrictions on the use of others' work, such as images and music downloaded from the internet. When incorporating the work of others, students attribute the work. At this level, students could give attribution by including credits or links directly in their programs, code comments, or separate project pages. For example, when making a program to model the life cycle of a butterfly, students could modify and reuse an existing program that describes the life cycle of a frog. Based on their research, students could identify and use Creative Commons-licensed or public domain images and sounds of caterpillars and butterflies. Students give attribution by properly citing the source of the original piece as necessary (CA NGSS: 3-LS-1-1) (CA CCSS for ELA/Literacy W.3.8, W.4.8, W.5.8). Alternatively, when creating a program explaining the structure of the United States government, students find Creative Commons-licensed or public domain images to represent the three branches of government and attribute ownership of the images appropriately. If students find and incorporate an audio file of a group playing part of the national anthem, they appropriately give attribution on the project page (HSS.3.4.4). 66 | California K–12 Computer Science Standards California Department of Education

3-5.AP.17 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.AP.17 Test

and debug a program or algorithm to ensure it accomplishes the intended task. Algorithms & Programming Program Development Testing 6.2 Descriptive Statement Programs do not always run properly. Students need to understand how to test and make necessary corrections to their programs to ensure they run properly. Students successfully identify and fix errors in (debug) their programs and programs created by others. Debugging strategies at this level may include testing to determine the first place the solution is in error and fixing accordingly, leaving “breadcrumbs” in a program, and soliciting assistance from peers and online resources. For example, when students are developing a program to control the movement of a robot in a confined space, students test various inputs that control movement of the robot to make sure it behaves as intended (e.g., if an input would cause the robot to move past a wall of the confined space, it should not move at all) (CA NGSS: 3-5-ETS1-3). Additionally, students could test and debug an algorithm by tracing the inputs and outputs on a whiteboard. When noticing “bugs” (errors), students could identify what was supposed to happen and step through the algorithm to locate and then correct the error. California Department of Education California K–12 Computer Science Standards | 67

3-5.AP.18 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.AP.18 Perform different roles when collaborating with peers during the design, implementation, and review stages of program development. Algorithms & Programming Program Development Collaborating 2.2 Descriptive Statement Collaborative computing is the process of creating computational artifacts by working in pairs or on teams. It involves asking for the contributions and feedback of others. Effective collaboration can often lead to better outcomes than working independently. With teacher guidance, students take turns in different roles during program development, such as driver, navigator, notetaker, facilitator, and debugger, as they design and implement their program. For example, while taking on different roles during program development, students could create and maintain a journal about their experiences working collaboratively (CA CCSS for ELA/Literacy W.3.10, W.4.10, W.5.10) (CA NGSS: 3-5-ETS1-2). 68 | California K–12 Computer Science Standards California Department of Education

3-5.AP.19 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.AP.19 Describe choices made during program development using code comments, presentations, and demonstrations. Algorithms & Programming Program Development Communicating 7.2 Descriptive Statement People communicate about their code to help others understand and use their programs. Explaining one’s design choices gives others a better understanding of one’s work. Students may explain their step-by-step process of creating a program in a presentation or demonstration of their personal code journals. They describe how comments within code organize thought and process during the development of the program. For example, students could describe the decision to have the score in a game flash when it can be rounded to 100 by writing a comment in the code (CA CCSS for Mathematics 3.NBT.1). Alternatively, students could present their overall program development experience and justify choices made by using storyboards, annotated images, videos, and/or journal entries (CA CCSS for ELA/Literacy SL.3.4, SL.4.4, SL.5.4, SL.3.5, SL.4.5, SL.5.5) (CA NGSS: 3-5-ETS1-1, 3.5-ETS1-2, 3.5-ETS1-3). California Department of Education California K–12 Computer Science Standards | 69

3-5.IC.20 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.IC.20 Discuss computing technologies that have changed the world, and express how those technologies

influence, and are influenced by, cultural practices. Impacts of Computing Culture Computational Problems 3.1 Descriptive Statement New computing technologies are created and existing technologies are modified for many reasons, including to increase their benefits, decrease their risks, and meet societal needs. Students, with guidance from their teacher, discuss topics that relate to the history of computing technologies and changes in the world due to these technologies. Topics could be based on current news content, such as robotics, wireless internet, mobile computing devices, GPS systems, wearable computing, and how social media has influenced social and political changes. For example, students could conduct research in computing technologies that impact daily life such as self-driving cars. They engage in a collaborative discussion describing impacts of these advancements (e.g., self-driving cars could reduce crashes and decrease traffic, but there is a cost barrier to purchasing them) (CA CCSS for ELA/Literacy W.3.7, W.4.7, W.5.7, SL.3.1, SL.4.1, SL.5.1). Alternatively, students could discuss how technological advancements affected the entertainment industry and then compare and contrast the impacts on audiences. For instance, people with access to high-speed internet may be able to choose to utilize streaming media (which may cost less than traditional media options), but those in rural areas may not have the same access and be able to reap those benefits (VAPA Theatre Arts 4.3.2, 4.4.2). 70 | California K–12 Computer Science Standards California Department of Education

3-5.IC.21 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.IC.21 Propose ways to improve the accessibility and usability of technology products for the diverse needs and wants of users. Impacts of Computing Culture Inclusion 1.2 Descriptive Statement The development and modification of computing technology is driven by people’s needs and wants and can affect groups differently. Students anticipate the needs and wants of diverse end users and propose ways to improve access and usability of technology, with consideration of potential perspectives of users with different backgrounds, ability levels, points of view, and disabilities. For example, students could research a wide variety of disabilities that would limit the use of traditional computational tools for the creation of multimedia artifacts, including digital images, songs, and videos. Students could then brainstorm and propose new software that would allow students that are limited by the disabilities to create similar artifacts in new ways (e.g., graphical display of music for the deaf, the sonification of images for visually impaired students, voice input for those that are unable to use traditional input like the mouse and the keyboard) (CA CCSS for ELA/Literacy W.3.7, W.4.7, W.5.7). Alternatively, as they anticipate unique user needs, students may consider using both speech and text to convey information in a game. They may also wish to vary the types of programs they create, knowing that not everyone shares their own tastes (CA NGSS: 3-5-ETS1-1, 3-5-ETS1-2, 3-5-ETS1-3). California Department of Education California K–12 Computer Science Standards | 71

3-5.IC.22 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.IC.22 Seek and explain the impact of diverse perspectives for the purpose of improving computational artifacts. Impacts of Computing Social Interactions Inclusion 1.1 Descriptive Statement Computing technologies enable global collaboration and sharing of ideas. Students solicit feedback from a diverse group of users and creators and explain how this input improves their computational artifacts. For example, students could seek feedback from classmates via user surveys, in order to create an idea and then make a claim as to how to improve the overall structure and function of their computational artifact. Using the feedback students could write an opinion piece supporting their claim (CA CCSS for ELA/Literacy W.3.1,

W.4.1, W.5.1). Alternatively, with guidance from their teacher, students could use video conferencing tools, shared documents, or other online collaborative spaces, such as blogs, wikis, forums, or website comments, to gather and synthesize feedback from individuals and groups about programming projects (CA CCSS for ELA/Literacy SL.3.1, SL.4.1, SL.5.1). 72 | California K–12 Computer Science Standards
California Department of Education

3-5.IC.23 Grade Standard Identifier Standard Framework Alignment: Concept Framework Alignment: Subconcept Framework Alignment: Practice(s) Framework Alignment: Subpractice(s) 3–5 3-5.IC.23 Describe reasons creators might limit the use of their work. Impacts of Computing Safety Law & Ethics Communicating 7.3 Descriptive Statement Ethical complications arise from the opportunities provided by computing. With the ease of sending and receiving copies of media on the internet, in formats such as video, photos, and music, students consider the opportunities for unauthorized use, such as online piracy and disregard of copyrights. The license of a downloaded image or audio file may restrict modification, require attribution, or prohibit use entirely. For example, students could take part in a collaborative discussion regarding reasons why musicians who sell their songs in digital format choose to license their work so that they can earn money for their creative efforts. If others share the songs without paying for them, the musicians do not benefit financially and may struggle to produce music in the future (CA CCSS for ELA/Literacy SL.3.1, SL.4.1, SL.5.1). Alternatively, students could review the rights and reproduction guidelines for digital artifacts on a publicly accessible media source. They could then state an opinion with reasons they believe these guidelines are in place (CA CCSS for ELA/Literacy W.3.1, W.4.1, W.5.1) California Department of Education California K–12 Computer Science Standards | 73