## **Foreword**

This document serves as a guideline only. Teachers are encouraged to explore Project lesson plans, platform content and learning journals themselves to figure out how to integrate each Project into their curriculum units. Although Projects offer a complete experience with their own self-contained content, we recommend that teachers look for areas where they can incorporate additional content that makes use of their own unique experience, knowledge and classroom learning objectives. If your curriculum is not yet included in one of these documents, that does not mean that this Project will not be suitable for your class. We recommend all teachers explore the Project content themselves before deciding how to integrate it into their curriculum as they know the requirements of their students and the topics they will find engaging better than anyone.

## Table Key

The contents of this Project are well suited to delivering this aspect of the curriculum.

As a teacher, you are free to skip over or de-emphasise content if this part of the curriculum is not something that you want to focus on. The contents of this Project can deliver part of this aspect of the curriculum.

This Project may require additional activities or discussions from the teacher to highlight key concepts of this aspect of the curriculum. The contents of this Project do not explicitly cover this aspect of the curriculum.

You can still use the simulations as a base for delivering your own content if you want to incorporate this aspect of the curriculum into the Project.

## USA - California

## Progression of California K-12 Computer Science Standards

	Grades K-2 Core	Grades 3–5 Core	Grades 6–8 Core	Grades 9–12 Core	Grades 9–12 Specialty
Computing Systems Devices	K-2.CS.1 Select and operate computing devices that perform a variety of tasks accurately and quickly based on user needs and preferences. (P1.1)	3-5.CS.1 Describe how computing devices connect to other components to form a system. (P7.2)	6-8.CS.1 Design modifications to computing devices in order to improve the ways users interact with the devices. (P1.2, P3.3)	9-12.CS.1 Describe ways in which abstractions hide the underlying implementation details of computing systems to simplify user experiences. (P4.1)	9-12S.CS.1 Illustrate ways computing systems implement logic through hardware components. (P4.4, P7.2)
Systems Hardware & Software	K-2.CS.2 Explain the functions of common hardware and software components of computing systems. (P7.2)	computer hardware and software work together as a	6-8.CS.2 Design a project that combines hardware and software components to collect and exchange data. (P5.1)	abstraction and interactions between application	9-12S.CS.2 Categorize and describe the different functions of operating system software. (P7.2)
Systems Troubleshooting	K-2.CS.3 Describe basic hardware and software problems using accurate terminology. (P6.2, P7.2)	solutions to solve simple hardware and software problems using common	strategies to identify and resolve hardware and	9-12.CS.3 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. (P6.2)	n/a
Internet Network Communication	K-2.NI.4 Model and describe how people connect to other people, places, information and ideas through a network. (P4.4)	into smaller pieces,	data across networks and the Internet. (P4.4)	functionality. (P4.1) 9-12.NI.5 Describe the design characteristics of the	scalability and reliability of networks, by describing the

		at the destination. (P4.4)			9-12S.NI.4 Explain how the characteristics of the Internet influence the systems developed on it. (P7.2)
Network & The Internet Cybersecurity	K-2.NI.5 Explain why people use passwords. (P7.2)	measures for protecting	6-8.NI.5 Explain potential security threats and security measures to mitigate threats. (P3.1, P3.3)	9-12.NI.6 Compare and contrast security measures to address various security threats. (P7.2)	9-12S.NI.5 Develop solutions to security threats. (P5.3)
	K-2.NI.6 Create patterns to communicate a message. (P4.4)	protect information from unauthorized access. (P4.4)	6-8.NI.6 Apply multiple methods of information protection to model the secure transmission of information. (P4.4)	9-12.NI.7 Compare and contrast cryptographic techniques to model the secure transmission of information. (P3.3, P4.4)	9-12S.NI.6 Analyze cryptographic techniques to model the secure transmission of information. (P3.3, P4.2)
Data & Analysis Storage	K-2.DA.7 Store, copy, search, retrieve, modify, and delete information using a computing device, and define the information stored as data. (P4.2)	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. (P4.2)	6-8.DA.7 Represent data in multiple ways. (P4.4)	9-12.DA.8 Translate between different representations of data abstractions of real-world phenomena, such as characters, numbers, and images. (P4.1) 9-12.DA.9 Describe tradeoffs associated with how data elements are organized and stored. (P3.3)	n/a
Data & Analysis Collection, Visualization, & Transformation	K-2.DA.8 Collect and present data in various visual formats. (P4.4, P7.1)	present collected data visually to highlight	6-8.DA.8 Collect data using computational tools and transform the data to make it more useful. (P7.1)	9-12.DA.10 Create data visualizations to help others better understand real-world phenomena. (P5.2)	9-12S.DA.7 Select and use data collection tools and techniques to generate data sets. (P7.1) 9-12S.DA.8 Use data analysis tools and techniques to identify patterns in data

					representing complex systems. (P4.1, P7.1)
Inference & Models	K-2.DA.9 Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions. (P4.1)	3-5.DA.9 Use data to highlight and/or propose relationships, predict outcomes, or communicate ideas. (P7.1)	6-8.DA.9 Test and analyze the effects of changing variables while using computational models. (P4.4, P6.1)	9-12.DA.11 Refine computational models to better represent the relationships among different elements of data collected from a phenomenon or process. (P4.4, P6.3)	9-12S.DA.9 Evaluate the ability of models and simulations to test and support the refinement of hypotheses. (P4.4)
Programming	K-2.AP.10 Model daily processes by creating and following algorithms to complete tasks. (P3.2, P4.4)	3-5.AP.10 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. (P3.3, P6.3)		9-12.AP.12 Design algorithms to solve computational problems using a combination of original and existing algorithms. (P4.2, P5.1)	9-12S.AP.10 Describe how artificial intelligence drives many software and physical systems. (P3.1, P7.2) 9-12S.AP.11 Implement an algorithm that uses artificial intelligence to overcome a simple challenge. (P3.1, P5.3) 9-12S.AP.12 Implement searching and sorting algorithms to solve computational problems. (P4.2, P5.2) 9-12S.AP.13 Evaluate algorithms in terms of their efficiency. (P3.3)
_	K-2.AP.11 Model the way programs store data. (P4.4)	3-5.AP.11 Create programs that use variables to store and modify data. (P5.2)	6-8.AP.11 Create clearly named variables that store data, and perform operations on their contents. (P5.1, P5.2)	9-12.AP.13 Create more generalized computational solutions using collections instead of repeatedly using simple variables. (P4.1)	9-12S.AP.14 Compare and contrast fundamental data structures and their uses.

Algorithms & Programming Control	K-2.AP.12 Create programs with sequences of commands and simple loops, to express ideas or	that include events, loops,	iteratively develop programs	and develop computational	9-12S.AP.15 Demonstrate the flow of execution of a recursive algorithm.(P3.2, P7.2)
	address a problem. (P5.2)		compound conditions. (P5.1, P5.2)	address a societal issue by using events to initiate instructions. (P5.1, P5.2, P5.3) 9-12.AP.14 Justify the selection of specific control structures by identifying tradeoffs associated with implementation, readability, and performance. (P5.2)	
Algorithms & Programming Modularity	K-2.AP.13 Decompose the steps needed to solve a problem into a sequence of instructions. (P3.2)	may themselves be decomposed. (P3.2) 3-5.AP.14 Create programs by incorporating smaller portions of existing	into parts to facilitate the design, implementation, and review of programs. (P3.2) 6-8.AP.14 Create procedures with parameters to organize code and make it easier to reuse. (P4.1, P4.3)	constructs such as	9-12S.AP.16 Analyze a large-scale computational problem and identify generalizable patterns or problem components that can be applied to a solution. (P3.2, P4.2) 9-12S.AP.17 Construct solutions to problems using student-created components, such as procedures, modules, and/or objects. (P4.3, P5.2) 9-12S.AP.18 Demonstrate code reuse by creating programming solutions using libraries and APIs. (P4.2, P5.3, P6.2)

Algorithms & K-2.AP.14 Develop plans that 3-5.AP.15 Use an iterative 6-8.AP.15	eek and 9-12.AP.18 Systematically 9-12S.AP.19 Plan and develop
IDragramana Idaaariba a program/a Inraaaaa ta plan and dayalan linaarnara	, , , , , , , , , , , , , , , , , , , ,
Programming describe a program's process to plan and develop incorpora	
	nbers and users to audiences by incorporating audiences using a specific
	lution that meets   feedback from users. (P1.1,   software life cycle process.
	s. (P1.1, P2.3) P5.1) (P2.2, P2.3, P5.2)
P5.1)	9-12S.AP.20 Develop
	programs for multiple
	computing platforms. (P5.2)
	9-12S.AP.26 Compare
	multiple programming
	languages, and discuss how
	their features make them
	suitable for solving different
	types of problems. (P7.2)
K-2.AP.15 Give attribution 3-5.AP.16 Observe 6-8.AP.16	ncorporate 9-12.AP.19 Explain the 9-12S.AP.23 Modify an
when using the ideas and intellectual property rights existing c	de, media, and limitations of licenses that existing program to add
creations of others while and give appropriate libraries in	to original restrict use of additional functionality and
developing programs. (P7.3) attribution when creating, programs	and give computational artifacts discuss intended and
remixing, or combining attribution	. (P4.2, P5.2, P7.3) when using resources such unintended implications.
programs. (P5.2, P7.3)	as libraries. (P7.3) (P4.2, P5.3)
K-2.AP.16 Debug errors in an 3-5.AP.17 Test and debug a 6-8.AP.17	ystematically test 9-12.AP.20 Iteratively 9-12S.AP.21 Identify and fix
algorithm or program that program or algorithm to and refine	programs using a evaluate and refine a security issues that might
includes sequences and ensure it accomplishes the range of t	est cases. (P6.1) computational artifact to compromise computer
simple loops. (P6.2) intended task. (P6.2)	enhance its performance, programs. (P6.2)
	reliability, usability, and 9-12S.AP.22 Develop and use
	accessibility. (P6.3) a series of test cases to
	verify that a program
	performs according to its
	design specifications. (P6.1)
3-5.AP.18 Perform different 6-8.AP.18	vistribute tasks 9-12.AP.21 Design and 9-12S.AP.25 Use version
roles when collaborating and main	ain a project develop computational control systems, integrated

		with peers during the design, implementation, and review stages of program development. (P2.2)	collaboratively developing	roles using collaborative tools. (P2.4)	development environments (IDEs), and collaborative tools and practices (e.g., code documentation) while developing software within a group. (P2.4, P5.2)
	K-2.AP.17 Describe the steps taken and choices made during the iterative process of program development.  (P7.2)	made during program development using code	6-8.AP.19 Document programs in order to make them easier to use, read, test, and debug. (P7.2)	decisions made during the design process using text,	9-12S.AP.24 Evaluate key qualities of a program through a process such as a code review. (P6.3)
Impacts of Computing Culture	K-2.IC.18 Compare how people lived and worked before and after the adoption of new computing technologies. (P3.1)	technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices. (P3.1) 3-5.IC.21 Propose ways to	6-8.IC.21 Discuss issues of bias and accessibility in the design of existing technologies. (P1.2)	computing impacts personal, ethical, social, economic, and cultural practices. (P1.2, P3.1) 9-12.IC.24 Identify impacts of bias and equity deficit on design and implementation of computational artifacts and apply appropriate processes for evaluating issues of bias. (P1.2) 9-12.IC.25 Demonstrate ways a given algorithm applies to problems across disciplines. (P3.1)	9-12S.IC.28 Evaluate how computational innovations that have revolutionized aspects of our culture might evolve. (P7.2) 9-12S.IC.29 Evaluate the impact of equity, access, and influence on the

				potential impacts and implications of emerging technologies on larger social, economic, and political structures, with evidence from credible sources. (P7.2)	
Impacts of Computing Social Interactions	' '	'	creating a computational	9-12.IC.27 Use collaboration tools and methods to increase connectivity with people of different cultures and careers. (P2.4)	n/a
Impacts of Computing Safety, Law, & Ethics	K-2.IC.20 Describe approaches and rationales for keeping login information private, and for logging off of devices appropriately. (P3.1)	creators might limit the use of their work. (P7.3)	associated with licenses for computational artifacts to balance the protection of the creators' rights and the ability for others to use and modify the artifacts. (P7.3) 6-8.IC.24 Compare tradeoffs between allowing information to be public and	data through automated	9-12S.IC.30 Debate laws and regulations that impact the development and use of software. (P7.2)