

Coding Robots CSTA Alignment

2-A	Framework Concept: Algorithms and Programming	Framework Practice	Lessons
2-A-2-1	Solicit and integrate peer feedback as appropriate to develop or refine a program.	Collaborating	1.1, 2.2, 2.3, 3.1, 4.3, 4.5, 5.5, 6.1, 7.3, 7.4, 7.5, 8.3, 8.5, 9.1, 9.3
2-A-7-2	Compare different algorithms that may be used to solve the same problem in terms of their speed, clarity, and size (e.g., different algorithms solve the same problem, but one might be faster than the other).	Communicating about Computing	1.1, 2.1, 2.2, 2.3, 2.4, 2.5, 3.1, 3.2, 4.2, 4.3, 4.4, 4.5, 5.1, 5.2, 5.3, 5.4, 5.5, 6.1, 6.2, 6.3, 6.4, 6.5, 7.1, 7.2, 7.3, 7.4, 7.5, 8.2, 8.3, 8.4, 8.5, 9.1, 9.3
2-A-7-4	Interpret the flow of execution of algorithms and predict their outcomes. [Clarification: Algorithms can be expressed using natural language, flow and control diagrams, comments within code, and pseudocode.	Communicating about Computing	1.2, 1.3, 1.4, 1.5, 2.1, 2.2, 2.3, 2.4, 2.5, 3.1, 3.2, 3.3, 3.4, 3.5, 4.1, 4.2, 4.3, 4.4, 4.5, 5.1, 5.2, 5.3, 5.4, 5.5, 6.1, 6.2, 6.3, 6.4, 6.5, 7.1, 7.2, 7.3, 7.4, 7.5, 8.1, 8.2, 8.3, 8.4, 8.5, 9.1, 9.3
2-A-5-6	Develop programs, both independently and collaboratively, that include sequences with nested loops and multiple branches	Creating Computational Artifacts	1.5, 2.4, 2.5, 3.2, 4.1, 4.2, 4.3, 4.4, 4.5, 5.1, 5.2, 5.4, 5.5, 6.2, 6.3, 6.4, 6.5, 7.3, 7.4, 7.5, 8.1, 8.2, 8.3, 8.4, 8.5, 9.3
2-A-5-7	Create variables that represent different types of data and manipulate their values	Creating Computational Artifacts	2.4, 4.1, 4.2, 4.4, 5.2, 5.3, 5.4, 5.5, 6.2, 7.1, 7.2, 7.3, 7.4, 7.5, 8.1, 8.2, 8.3, 8.4, 8.5, 9.3
2-A-4-8	Define and use procedures that hide the complexity of a task and can be reused to solve	Developing and Using Abstractions	1.3, 1.4, 1.5, 2.1, 2.2, 2.3, 2.4, 2.5, 3.1, 3.2, 3.3, 3.4, 3.5, 4.1, 4.2,

	similar tasks. [Clarification: Students use and modify, but do not necessarily create, procedures with parameters.]		4.3, 4.4, 4.5, 5.1, 5.2, 5.3, 5.4, 5.5, 6.1, 6.2, 6.3, 6.4, 6.5, 7.1, 7.2, 7.3, 7.4, 7.5, 8.1, 8.2, 8.3, 8.4, 8.5, 9.3
2-A-3-9	Decompose a problem into parts and create solutions for each part	Recognizing and Defining Computational Problems	1.3, 1.4, 1.5, 2.1, 2.2, 2.3, 2.4, 2.5, 3.1, 3.2, 3.3, 3.4, 3.5, 4.1, 4.2, 4.3, 4.4, 4.5, 5.1, 5.2, 5.3, 5.4, 5.5, 6.1, 6.2, 6.3, 6.4, 6.5, 7.1, 7.2, 7.3, 7.4, 7.5, 8.1, 8.2, 8.3, 8.4, 8.5, 9.1, 9.3
2-A-6-10	Use an iterative design process (e.g., define the problem, generate ideas, build, test, and improve solutions) to solve problems, both independently and collaboratively	Testing and Refining	1.3, 1.4, 2.1, 2.2, 2.3, 2.4, 2.5, 3.1, 3.2, 3.4, 3.5, 4.1, 4.2, 4.3, 4.4, 4.5, 5.1, 5.2, 5.3, 5.4, 5.5, 6.1, 6.2, 6.3, 6.4, 6.5, 7.1, 7.2, 7.3, 7.4, 7.5, 8.1, 8.2, 8.3, 8.4, 8.5, 9.1, 9.3

2-C	Framework Concept: Computing Systems	Framework Practice	Lessons
2-C-7-11	Justify the hardware and software chosen to accomplish a task	Communicating about Computing	1.1, 1.4, 1.5, 2.2, 2.3, 2.4, 3.1, 3.2, 3.3, 3.5, 4.3, 4.4, 4.5, 5.1, 5.2, 5.3, 5.4, 5.5, 6.1, 6.2, 6.3, 6.4, 7.1, 7.5, 8.4, 8.5, 9.1, 9.3
2-C-4-12	Analyze the relationship between a device's computational components and its capabilities	Developing and Using Abstractions	1.1, 1.4, 1.5, 2.1, 2.2, 2.3, 2.4, 3.1, 3.2, 3.3, 4.3, 4.4, 5.1, 5.3, 5.4, 6.1, 6.2, 6.3, 6.4, 7.1, 7.5, 8.5
2-C-6-13	Use a systematic process to	Testing and	1.1, 1.4, 1.5, 2.2, 4.3,

	identify the source of a problem within individual and connected devices	Refining	5.2, 6.1, 6.3, 6.4, 7.1
--	--	----------	-------------------------

2-D	Framework Concept:Data and Analysis	Framework Practice	Lessons
2-D-7-15	Explain the processes used to collect, transform, and analyze data to solve a problem using computational tools	Communicating about Computing	2.3, 2.4, 2.5, 3.2, 3.4, 3.5, 4.2, 5.2, 5.3, 6.1, 6.3, 7.3

2-I	Framework Concept: Impacts of Computing	Framework Practice	Lessons
2-I-7-19	Explain how computer science fosters innovation and enhances nearly all careers and disciplines.	Communicating about Computing	1.2
2-I-1-20	Provide examples of how computational artifacts and devices impact health and wellbeing, both positively and negatively.	Fostering an Inclusive Computing Culture	1.2