BOOTSTRAP
www.bootstrapworld.org

AP CSP Performance Task Learning Objectives mapped to Bootstrap:Reactive

2.2.1	Develop an abstraction when writing a program or creating other computational artifacts. [P2]	Bootstrap:Reactive builds off of Bootstrap:Algebra's core idea of functions as abstractions over repeated computations. In Bootstrap:Reactive's final project, students write functions to consume and produce structured data as a way to decompose large problems into sub-problems to be designed, implemented, and tested independently.
4.1.1	Develop an algorithm for implementation in a program. [P2]	Bootstrap emphasizes computation, particularly the idea that computations are reusable with different inputs. Through their games and in-class exercises, students see various applications for reusable computations in practical settings. In Bootstrap:Reactive, students learn about how the event-loop driving a video game works, and develop their own series of steps to implement the mechanics of a game or animation of their own design. For example, in order to control boundary detection or collisions between characters, students must design and program an algorithm to implement those features.

4.1.2	Express an algorithm in a language. [P5]	Bootstrap students consider many algorithms throughout the material. They are confronted with multiple ways to reach a solution, asked to generalize algorithms so they can be re-used for other problems, and given opportunities to discuss the tradeoffs between different approaches, before implementing what they've learned by programming a unique game or animation.
5.1.1*	Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. [P2]	Students are given the freedom to design and implement a complete game or animation of their own design.
5.1.2**	Develop a correct program to solve problems. [P2]	Bootstrap students build programs that are meant to solve problems they care about. Past examples include making "picture books" to teach autistic siblings how to identify mood and emotion, or videogames designed to combat diabetes by encouraging healthy eating habits or to save the environment through ocean conservation, etc.
5.2.1	Explain how programs implement algorithms. [P3]	Bootstrap emphasizes computation, particularly the idea that computations are reusable with different inputs. Through their projects and in-class exercises, students see various applications for reusable computations in practical settings.

5.3.1	Use abstraction to manage complexity in programs. [P3]	Bootstrap:Reactive builds off of Bootstrap:Algebra's core idea of functions as abstractions over repeated computations. In Bootstrap:Reactive's final project, students write functions to consume and produce structured data as a way to decompose large problems into sub-problems to be designed, implemented, and tested independently. This decomposition is extended further as students begin writing helper functions to manage repeated code inside of larger functions, as well as using nested structures to manage the complexity of working with large data structures.
5.4.1	Evaluate the correctness of a program. [P4]	Bootstrap teaches students to develop programs and solve problems through a concrete sequence of steps called the Design Recipe. Summarizing problem statements, writing test cases, designing functions around data, coding, and testing are all explicit steps in this recipe. Bootstrap:Reactive emphasizes testing throughout the course, and extends the steps of the Design Recipe learned in Bootstrap:Algebra into richer, structured data. Students are taught to evaluate the correctness of a program not by the behavior of the running program, but by the results of comprehensive unit tests written for each function in the program.

5.5.1	Employ appropriate mathematical and logical concepts in programming. [P1]	Bootstrap puts mathematical and logical functions in the context of programming video games. Bootstrap:Reactive covers functions, variables, logic, and conditionals (piecewise functions) as part of the process to create a rich game or animation.

* Learning Objective 5.1.1 is assessed if students decide to create a program for the purpose of self-expression.

** Learning Objective 5.1.2 is assessed if students decide to create a program to solve a problem.