BJC Outline of Units

Unit 1: Introduction to Programming

Students develop an interactive game they can install on their phones, generate a conversation between animated characters, create abstract art, and explore storytelling animation through sprite interaction (**AAP**); and in doing so, they learn to use pair programming and to create program documentation. (**CRD**) Students also investigate legal and ethical issues that arise in computing—especially with regard to data collection and privacy (**IOC**).

Unit 2: Abstraction

Students implement an algorithm for a guessing game using local and global variables; use abstract data types and list traversal to build a quiz app; create predicates to filter lists in order to solve a crossword puzzle; and use the modulus function and a higher order function to code mathematical functions (**AAP**). Students also investigate the history, purpose, laws, evolution, and enforcement of copyright (**IOC**).

Unit 3: Data Structures

Students explore complexity in a variety of contexts (maze navigation, fractal art, tic-tac-toe); use nested abstract data types and data I/O to develop a contact list app; and consider the beneficial and harmful impacts of robots and AI (**CRD**, **AAP**, **IOC**).

Practice AP Create Task

Students create a project of their own choosing as practice for the AP Create Task. They select and use a development process, plan and code their program, test their program for errors, write about their development process, and acknowledge any code developed by other people (**CRD**).

Unit 4: How the Internet Works

Students learn about how the Internet works, the benefits and vulnerabilities of fault-tolerant systems; cybersecurity practices such as public key encryption and individual level practices and software to keep data safe; digital data representation including binary representation; compression algorithms (**CSN**, **IOC**, **DAT**). They also consider the impact of the Internet on human communication and the workplace (**CRD**, **IOC**).

Unit 5: Algorithms and Simulations

Students learn about program efficiency through exploration of the binary and linear search algorithms; learn about sequential, parallel, and distributed computing and determine the contexts in which each are most useful; consider the contexts in which simulation is useful and

implement a simple simulation; use Snap*!* data tools to generate knowledge from data (**AAP**, **CSN**, **DAT**).

AP Create Task

Students complete the *AP Create Task* (12 hours in class).

Note: Units 1-5, the Practice Create Task, and the Create Task cover all of the 2020 AP CSP curriculum framework. Units 6-8 focus on the abstraction hierarchy of how computers work and recursion, a beautiful and powerful CS idea that goes beyond the AP CSP Framework and exam.

Unit 6: How Computers Work

Building on their understanding of abstraction and the way computers store data, students learn about the computer system abstraction hierarchy, with application software on top and transistors at the bottom.

Unit 7: Fractals and Recursion

Students deepen their experience with recursion and functional programming through drawing projects that use recursive commands, mainly fractals.

Unit 8: Recursive Functions

Students extend their understanding of abstraction and recursion through exploration of recursive functions: sorting lists (both selection sort and partition sort), Pascal's triangle, converting numbers to and from binary, finding the subsets of a set, and building several higher order functions from scratch.