Concept	Subconcept	Level 3B (Ages 16-18)	Course	Current Status
Computing Systems	Hardware & Software	3B-CS-01 Categorize the roles of operating system software. Practice(s): Communicating About Computing: 7.2	FCCS	Already Meeting
Computing Systems	Troubleshooting	3B-CS-02 Illustrate ways computing systems implement logic, input, and output through hardware components. Practice(s): Communicating About Computing: 7.2	FCCS	Already Meeting
Networks & The Internet	Network Communication & Organization	3B-NI-03 Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology). Practice(s): Communicating About Computing: 7.2	FCCS	Progressing
Networks & The Internet	Cybersecurity	3B-NI-04 Compare ways software developers protect devices and information from unauthorized access. Practice(s): Communicating About Computing: 7.2	FCCS	Progressing
Data & Analysis	Collection, Visualization, & Transformation	3B-DA-05 Use data analysis tools and techniques to identify patterns in data representing complex systems. Practice(s): Developing and Using Abstractions: 4.1		Unmet
Data & Analysis	Collection, Visualization, & Transformation	3B-DA-06 Select data collection tools and techniques to generate data sets that support a claim or communicate information. Practice(s): Communicating About Computing: 7.2		Unmet
Data & Analysis	Inference & Models	3B-DA-07 Evaluate the ability of models and simulations to test and support the refinement of hypotheses. Practice(s): Developing and Using Abstractions: 4.4		Unmet
Algorithms & Programming	Algorithms	3B-AP-08 Describe how artificial intelligence drives many software and physical systems. Practice(s): Communicating About Computing: 7.2	FCCS	Progressing
Algorithms & Programming	Algorithms	3B-AP-09 Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem. Practice(s): Creating Computational Artifacts: 5.3	DSA	Already Meeting
Algorithms & Programming	Algorithms	3B-AP-10 Use and adapt classic algorithms to solve computational problems. Practice(s): Developing and Using Abstractions: 4.2	DSA	Progressing
Algorithms & Programming	Algorithms	3B-AP-11 Evaluate algorithms in terms of their efficiency, correctness, and clarity. Practice(s): Developing and Using Abstractions: 4.2	DSA	Already Meeting
Algorithms & Programming	Variables	3B-AP-12 Compare and contrast fundamental data structures and their uses. Practice(s): Developing and Using Abstractions: 4.2	DSA	Already Meeting
Algorithms & Programming	Control	3B-AP-13 Illustrate the flow of execution of a recursive algorithm. Practice(s): Recognizing and Defining Computational Problems: 3.2	DSA	Progressing

Concept	Subconcept	Level 3B (Ages 16-18)	Course	Current Status
Algorithms & Programming	Modularity	3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects. Practice(s): Creating Computational Artifacts: 5.2	DSA	Progressing
Algorithms & Programming	Modularity	3B-AP-15 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution. Practice(s): Developing and Using Abstractions: 4.1	DSA	Unmet
Algorithms & Programming	Modularity	3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs. Practice(s): Creating Computational Artifacts: 5.3	DSA	Unmet
Algorithms & Programming	Program Development	3B-AP-17 Plan and develop programs for broad audiences using a software life cycle process. Practice(s): Creating Computational Artifacts: 5.1		Unmet
Algorithms & Programming	Program Development	3B-AP-18 Explain security issues that might lead to compromised computer programs. Practice(s): Communicating About Computing: 7.2	FCCS	Progressing
Algorithms & Programming	Program Development	3B-AP-19 Develop programs for multiple computing platforms. Practice(s): Creating Computational Artifacts: 5.2		Unmet
Algorithms & Programming	Program Development	3B-AP-20 Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project. Practice(s): Collaborating Around Computing: 2.4		Unmet
Algorithms & Programming	Program Development	3B-AP-21 Develop and use a series of test cases to verify that a program performs according to its design specifications. Practice(s): Testing and Refining Computational Artifacts: 6.1	FOP	Progressing
Algorithms & Programming	Program Development	3B-AP-22 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality). Practice(s): Creating Computational Artifacts: 5.3	FOP	Unmet
Algorithms & Programming	Program Development	3B-AP-22 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality). Practice(s): Creating Computational Artifacts: 5.4	DSA	Unmet
Algorithms & Programming	Program Development	3B-AP-23 Evaluate key qualities of a program through a process such as a code review. Practice(s): Testing and Refining Computational Artifacts: 6.3		Unmet
Algorithms & Programming	Program Development	3B-AP-24 Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems. Practice(s): Communicating About Computing: 7.2	FOP and TLP	Progressing

Concept	Subconcept	Level 3B (Ages 16-18)	Course	Current Status
Impacts of Computing	Culture	3B-IC-25 Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society. Practice(s): Testing and Refining Computational Artifacts, Fostering an Inclusive Computing Culture: 6.1, 1.2		Unmet
Impacts of Computing	Culture	3B-IC-26 Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society. Practice(s): Fostering an Inclusive Computing Culture: 1.2	TLP	Progressing
Impacts of Computing	Culture	3B-IC-27 Predict how computational innovations that have revolutionized aspects of our culture might evolve. Practice(s): Communicating About Computing: 7.2	FCCS	Progressing
Impacts of Computing	Safety, Law, & Ethics	3B-IC-28 Debate laws and regulations that impact the development and use of software. Practice(s): Recognizing and Defining Computational Problems, Communicating About Computing: 3.3, 7.3	FCCS	Progressing