Crosswalk - CSTA Standards - Foundations of Computer Science For Teachers - Praxis 5652 Certification Prep

CSTA Standards	Foundations of CS for Teachers - Praxis 5652 Modules and Lessons

Level 3A: Grades 9-10 (Ages 14-16)

3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.	2A1: Abstraction_
3A-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers.	Module 5: A4 - Hardware/Software/Layers of Abstractions
3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	Module 3: Debugging Strategies
3A-NI-04 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.	5B1,2 and 3: Network Components, Functionality, Protocols
3A-NI-05 Give examples to illustrate how sensitive data can be affected by malware and other attacks.	Module 5: B4 - Network Security Strategies
3A-NI-06 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.	5B5: Five Pillars of Cybersecurity
3A-NI-07 Compare various security measures, considering tradeoffs between the usability and security of a computing system.	Module 5: B4 - Network Security Strategies
3A-NI-08 Explain tradeoffs when selecting and implementing cybersecurity recommendations.	5B5: Five Pillars of Cybersecurity
3A-DA-09 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.	2A3: Computer Number Bases
3A-DA-10 Evaluate the tradeoffs in how data elements are organized and where data is stored.	4B2: Data Storage and Management
3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	2A2: Developing Algorithms_
3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.	Module 3: A6 1D Arrays, Terms, Guided Practice
3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.	2A2: Developing Algorithms_

3A-AP-18 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.	Module 3: B1 - Procedures and Parameters
3A-AP-20 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.	3B8: Using Libraries and APIs
3A-AP-23 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.	2A4: Pseudocode, Flowcharts
3A-IC-24 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.	Module 1 - Impacts of Computing
3A-IC-28 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.	Module 1: B1 - Intellectual Property Issues
3A-IC-29 Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.	Module 1: B3 - Digital Privacy and Security
3A-IC-30 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.	Module 1: B2 - Ethics of Computing

Level 3B: Grades 11-12 (Ages 16-18)

3B-CS-01 Categorize the roles of operating system software.	5A1: Operating Systems/Hardware/Software
3B-CS-02 Illustrate ways computing systems implement logic, input, and output through hardware components.	5A1: Operating Systems/Hardware/Software
3B-NI-03 Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).	5B1,2 and 3: Network Components, Functionality, Protocols
3B-NI-04 Compare ways software developers protect devices and information from unauthorized access.	Module 1: B3 - Digital Privacy and Security
3B-AP-10 Use and adapt classic algorithms to solve computational problems.	2B2: Searching and Sorting Algorithms
3B-AP-11 Evaluate algorithms in terms of their efficiency, correctness, and clarity.	Module 2: B1 - Space/Time Limitations and Heuristics
3B-AP-12 Compare and contrast fundamental data structures and their uses.	3A6: Part Two - Procedures, Parameters, Arrays, Lists, Data Structures
3B-AP-13 Illustrate the flow of execution of a recursive algorithm.	Module 2: B3 - Recursive Algorithms
3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.	Module 3: B1 - Procedures and Parameters
3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.	3B8: Using Libraries and APIs
3B-AP-17 Plan and develop programs for broad audiences using a software life cycle process.	3A3: Extensibility, Modifiability, Reusability_
3B-AP-18 Explain security issues that might lead to compromised computer programs.	Module 1: B3 - Digital Privacy and Security

3B-AP-24 Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.	3B11,12,14: Programming Language Concepts
3B-IC-26 Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.	1A3: Computing Innovations - Benefits and Tradeoffs
3B-IC-28 Debate laws and regulations that impact the development and use of software.	Module 1: B1 - Intellectual Property Issues