Exam: JavaScript Coding Specialist

Exam Link: https://knowledge-pillars.com/javascript-coding-specialist-certification/

Course: Computer Science II

TEKS Code & Standard	TEKS Description	Corresponding Python JavaScript Coding Specialist Exam Objective(s)	Alignment Description
§127.785(c)(1)(A)	Describe the function of a software development environment.	1.2 – JavaScript variables and variable declarations	JavaScript variable scoping and declarations mirror setup in IDEs and environments.
§127.785(c)(2)(B)	Design solutions using control structures such as if-then and loops.	3.2 – Loops and iteration statements	Loops and control flow statements are core to JavaScript, matching CSII logic structures.
§127.785(c)(3)(C)	Implement input validation and error handling in programs.	3.3 – Errors and exception handling	try/catch blocks directly map to input validation and handling exceptions.
§127.785(c)(4)(A)	Create modular programs with functions and parameters.	4 – Functions and function parameters	Functions with parameters and return values build modular code in both standards.
§127.785(c)(4)(B)	Use arrays and collections to manage data sets.	1.4 – JavaScript literals; 5.1 – Objects and arrays	JavaScript arrays, objects, and literals align with TEKS data structures.
§127.785(c)(5)(C)	Debug and test programs to ensure correctness and efficiency.	3.3 – Error handling; 2.2 – Expressions	Testing via expressions and error handling simulates debugging in student programs.
§127.785(c)(6)(A)	Understand and apply object-oriented programming principles.	5.4 – Inheritance; 5.3 – Creating new objects	OOP features such as constructors and inheritance support advanced CSII coding principles.
§127.785(c)(2)(A)	Design solutions that utilize data and apply coding logic.	2.1 – Types of operators and operator precedence	JavaScript operators and logical structures support data-driven logic building and calculations.
§127.785(c)(3)(A)	Use project management principles in software design.	6.5 – Interact with the Browser Object Model (BOM)	BOM operations and browser scripting simulate applied project environment interactions and requirements.
§127.785(c)(3)(B)	Use standard algorithms to process data.	5.7 – Math Object and built-in functions	Math methods and object utilities in JavaScript allow students to perform data processing and numeric analysis.
§127.785(c)(5)(A)	Identify and analyze user feedback to improve content and layout.	6.2 – Handle HTML events	Event handlers such as onClick and onChange help capture user interaction for feedback-based improvements.
§127.785(c)(5)(B)	Use version control systems to track changes and collaborate in software development.	6.1 – Construct the DOM tree; 6.3 – Output to HTML	DOM-based changes simulate document versioning, and code output logs changes to visual structure, useful for collaboration tracking.

Course: Computer Science III

TEKS Code & Standard	TEKS Description	Corresponding Python JavaScript Coding Specialist Exam Objective(s)	Alignment Description
§127.786(c)(1)(A)	Understand and apply advanced development environments.	1.2 – JavaScript variables and variable declarations	JavaScript variable scoping and declarations mirror setup in IDEs and environments.
§127.786(c)(2)(A)	Design solutions using structured data and logic.	3.2 – Loops and iteration statements	Loops and control flow statements are core to JavaScript, matching CSII logic structures.
§127.786(c)(2)(B)	Apply conditional and iterative logic to solve problems.	3.3 – Errors and exception handling	try/catch blocks directly map to input validation and handling exceptions.
§127.786(c)(3)(A)	Use project planning principles in software design.	4 – Functions and function parameters	Functions with parameters and return values build modular code in both standards.
§127.786(c)(3)(B)	Implement standard algorithms to solve data problems.	1.4 – JavaScript literals; 5.1 – Objects and arrays	JavaScript arrays, objects, and literals align with TEKS data structures.
§127.786(c)(3)(C)	Use exception handling to improve system resilience.	3.3 – Error handling; 2.2 – Expressions	Testing via expressions and error handling simulates debugging in student programs.
§127.786(c)(4)(A)	Create modular code using functions or methods.	5.4 – Inheritance; 5.3 – Creating new objects	OOP features such as constructors and inheritance support advanced CSII coding principles.