SOFTWARE PROGRAMMING & APP DESIGN 1

ACADEMIC STANDARDS

CATALINA FOOTHILLS SCHOOL DISTRICT

Approved by Governing Board on May 28, 2019

Catalina Foothills School District
Software Programming & App Design 1
Grades: 10-12

Software Programming & App Design 1 introduces students to the fundamental concepts and procedures of computer programming, which includes the basics of object-oriented programming languages. With a focus on computational thinking, students will solve problems through the design, execution, and refinement of solutions. Students will learn programming using the Swift programming language while demonstrating how these concepts can be applied to other programming languages. Students will utilize Apple's XCode software to create fully functional apps for iOS devices. Other topics include data storage and management (e.g., database applications that work with different programming languages); legal, ethical, and security issues related to informational technology; and the employability and leadership skills associated with business and industry in the field of computer science. The Software Programming & App Design program will prepare students for industry certification.

1. Problem-Solving and Critical Thinking	
SPAD1.1.1	Identify problems and use strategies and resources to innovate and/or devise plausible solutions.
SPAD1.1.2	Explain the process of decomposing a large programming problem into smaller, more manageable procedures.
SPAD1.1.3	Implement problem-solving and troubleshooting strategies applicable to software development (for example: debugging).
SPAD1.1.4	Take action or make decisions supported by evidence and reasoning.
SPAD1.1.5	Transfer knowledge/skills from one situation/context to another.
2. Legal and Ethical Issues	
SPAD1.2.1	Follow intellectual property rights including software licensing and software duplication [required: Digital Millennium Copyright Act (DMCA), software licensing, software duplication]. ○ compare and contrast open source and proprietary systems in relation to legal and ethical issues (required: data pricing, use of public and private networks, social networking, industry-related data, data piracy)
SPAD1.2.2	Compare and contrast issues and regulations affecting computers, other devices, the internet, and information privacy (required: HIPAA, COPPA, CISPA, FERPA, PCI, GDPR, data brokers).
3. Security Issues	
SPAD1.3.1	Identify common computer threats (required: viruses, phishing, suspicious email, social engineering, spoofing, identity theft, spamming).
SPAD1.3.2	Implement procedures to maintain data integrity and security (required: lock the screen, delete unrecognized emails, use trustworthy thumb drives, use approved software).
SPAD1.3.3	Describe methods for sanitizing user input to prevent issues (required: data validation).
SPAD1.3.4	Explain the CIA (confidentiality, integrity, and availability) triad.
4. Primitive Data Types and Strings	
SPAD1.4.1	Use primitive data types in writing programs.
SPAD1.4.2	Use strings in writing programs.
SPAD1.4.3	Research data types across industry relevant program languages (for example: Java, JavaScript, and Python).
5. Basic Computer Mathematics	
SPAD1.5.1	Apply basic mathematics to hardware (required: bits, bytes, kilobytes, megabytes, gigabytes, terabytes.

SPAD1.5.2	Use binary to decimal, decimal to hexadecimal, hexadecimal to decimal, binary to hexadecimal, and binary to hexadecimal conversions to solve hardware and software problems.
SPAD1.5.3	Apply basic mathematics operations and conversions to popular memory sizes (bits, bytes, kilobytes, etc.) and number systems in computing (binary to decimal, decimal to hexadecimal, hexadecimal to decimal, etc.).
SPAD1.5.4	Construct and apply mathematical operations within programs.
6. CONDITIONAL STATEMENTS/STRUCTURES	
SPAD1.6.1	Utilize conditional statements/structures in writing programs. ○ use conditionals to provide varied functionality based on user input
7. ITERATIVE STRUCTURES	
SPAD1.7.1	Utilize iterative structures in writing programs. ○ iterate through an array to perform a common action on each item in the collection
8. BASIC DATA STRUCTURES	
SPAD1.8.1	Compare and contrast multiple techniques for declaring, initializing, and modifying user-defined data types.
SPAD1.8.2	Create and use two-dimensional arrays to manage a collection of objects.
9. INTERNET PROTOCOLS AND OPERATIONS	
SPAD1.9.1	Describe Internet protocols and operations within appropriate context.
SPAD1.9.2	Identify performance issues (e.g. bandwidth, internet connection types, pages loading slowly, resolution and size of graphics).
10. CLIENT-SIDE INTERNET SOFTWARE	
SPAD1.10.1	Identify key components and functions of internet and web specialty browsers..
SPAD1.10.2	Select and use the most appropriate clients (for example: GitHub, Google Drive, Dropbox, JSFiddle, browser developer tools).
SPAD1.10.3	Analyze remote computing tools and services and their application.
11. PROGRAM ANALYSIS AND DESIGN	
SPAD1.11.1	Implement the steps in the System Development Life Cycle (SDLC) (required: planning, analysis, design, development, testing, implementation, maintenance). ○ develop requirements/specifications and a testing plan for a program (required: client requirements, automated testing, test procedures) ○ apply pseudocode or graphical representations to plan the structure of a program or module (required: flowcharting, white boarding)
SPAD1.11.2	Create and implement basic algorithms.
12. PROGRAM DEVELOPMENT	
SPAD1.12.1	Use an editor to create and modify code.
SPAD1.12.2	Identify correct input/output statements.
SPAD1.12.3	Choose the correct method of assigning input to variables including data sanitization.
SPAD1.12.4	Choose correct method of outputting data with formatting and escaping.
SPAD1.12.5	Differentiate between interpreted and compiled code (required: steps necessary to run executable code).
SPAD1.12.6	Apply industry standards in documentation (required: self-documenting code; function-level, program-level, user-level documentation).
SPAD1.12.7	Name identifiers and format code by applying conventions.
SPAD1.12.8	Demonstrate the use of *parameters* to pass data into program modules and return values from modules.
13. TESTING AND DEBUGGING	
SPAD1.13.1	Identify and categorize errors in program modules. ○ identify boundary cases and generate appropriate test data

SPAD1.13.2	Perform integration testing including tests within a program to protect execution from bad input or other run-time errors.
SPAD1.13.3	Perform different methods of debugging (required: hand-trace code or real time debugging tools).
14. Community Resource Utilization and Creation	
SPAD1.14.1	Use standard library functions.
SPAD1.14.2	Find and use third party libraries.
SPAD1.14.3	Explain and interact with an Application Program Interface (API).
15. Version Control Systems	
SPAD1.15.1	Explain the purpose of version control systems (Git, Mercurial).
SPAD1.15.2	Create a new repository. ○ add, push, and pull source code from repository ○ restore previous versions of code from the repository
16. User Design Principles of Websites and Applications	
SPAD1.16.1	Apply W3C standards and style conventions.
SPAD1.16.2	Construct web pages and applications that are compliant with ADA and sections 504 and 508 standards.
SPAD1.16.3	Explain the concept of responsive design and applications.
SPAD1.16.4	Employ graphics methods to create images at specified locations.
SPAD1.16.5	Discriminate between or among GUI objects for input and output of data.
17. User Design Principles of Websites and Applications	
SPAD1.17.1	Describe various locations for storing data (files, SQL DB, key-value store, graph DB, cloud).
SPAD1.17.2	Input/output data from a sequential file or database.
SPAD1.17.3	Demonstrate the proper use of database applications.
18. User Design Principles of Websites and Applications	
SPAD1.18.1	Make a distinction between an object and a class.
SPAD1.18.2	Create a user-defined class.
SPAD1.18.3	Read the state of an object by invoking accessor methods.
SPAD1.18.4	Change the state of an object by invoking a modifier method
SPAD1.18.5	Determine the requirements for constructing new objects by reading the documentation.
19. Run Time and Error Handling Techniques	
SPAD1.19.1	Explain types of runtime errors. ○ discriminate between runtime and compile time errors (for example: syntax errors)
SPAD1.19.2	Describe error handling strategies.
SPAD1.19.3	Handle unexpected return values.
SPAD1.19.4	Handle (catch) run time errors and take appropriate action.
20. Workplace Employability: Professionalism	
SPAD1.20.1	Demonstrate professionalism in the workplace (being on time, proper dress, courteousness).
SPAD1.20.2	Represent the school [or organization] in a positive manner, demonstrating the school's [or organization's] mission and core values.
SPAD1.20.3	Demonstrate respect for personal and professional boundaries (distinguish between personal and work-related matters).
SPAD1.20.4	Interact respectfully with others; act with integrity.
SPAD1.20.5	Produce high quality work that reflect professional pride and contributes to organizational success.
SPAD1.20.6	Take initiative to develop skills and improve work performance.
21. Workplace Employability: Communication (traditional and digital)	
SPAD1.21.1	Communicate effectively in preparation for a diverse work environment (required: style, format, and medium appropriate to audience/culture/generation, purpose and context; accuracy; use of

	appropriate technical/industry language; to resolve conflicts; address intergenerational differences/challenges; persuade others).
SPAD1.21.2	Use documentation (for example: itineraries and schedules) to plan and meet client needs.
SPAD1.21.3	Use appropriate technologies and social media to enhance or clarify communication.
SPAD1.21.4	Use a variety of interpersonal skills, including tone of voice and appropriate physical gestures (for example: eye contact, facing the speaker, active listening) during conversations and discussions to build positive rapport with others.
SPAD1.21.5	Pose and respond to questions, building upon others' ideas in order to enhance the discussion; clarify, verify, or challenge ideas and conclusions with diplomacy.
22. WORKPLACE EMPLOYABILITY: SELF-REGULATION	
SPAD1.22.1	Apply the skills and mindset of self-regulation to accomplish a task or project.
SPAD1.22.2	Select and use appropriate technologies to increase productivity.
SPAD1.22.3	Exercise initiative and leadership (for example: recognize and engage individual strengths, plan for unanticipated changes, pursue solutions/improvements).
23. WORKPLACE EMPLOYABILITY: COLLABORATION	
SPAD1.23.1	Take responsibility for any role on a team and accurately describe and perform the duties of each role, including leadership.
SPAD1.23.2	Integrate diverse ideas, opinions, and perspectives of the team and negotiate to reach workable solutions.
SPAD1.23.3	Prioritize and monitor individual and team progress toward goals, making sufficient corrections and adjustments when needed.
SPAD1.23.4	Submit high-quality products that meet specifications for assigned tasks.
SPAD1.23.5	Participate in shared note-taking as directed.