AP COMPUTER SCIENCE PRINCIPLES

LENGTH OF TIME: 90 minutes every other day for full year
GRADE LEVEL: 9-12

CSTA COMPUTER SCIENCE STANDARDS:

Category	CSTA Standard	Description
Algorithms & Programming	2-AP-10	Use flowcharts and/or pseudocode to address complex problems as algorithms.
Algorithms & Programming	2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.
Algorithms & Programming	2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
Algorithms & Programming	2-AP-14	Create procedures with parameters to organize code and make it easier to reuse.
Algorithms & Programming	2-AP-17	Systematically test and refine programs using a range of test cases.
Algorithms & Programming	2-AP-19	Document programs in order to make them easier to follow, test, and debug.
Algorithms & Programming	3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
Algorithms & Programming	3A-AP-14	Use lists to simplify solutions, generalizing computational problems instead of repeated use of simple variables.
Algorithms & Programming	3A-AP-15	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance and explain the benefits and drawbacks of choices made.
Algorithms & Programming	3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
Algorithms & Programming	3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
Algorithms & Programming	3A-AP-18	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
Algorithms & Programming	3A-AP-19	Systematically design and develop programs for broad audiences by incorporating feedback from users.
Algorithms & Programming	3A-AP-21	Evaluate and refine computational artifacts to make them more usable and accessible.
Algorithms & Programming	3A-AP-22	Design and develop computational artifacts working in team roles using collaborative tools.

Algorithms & Programming	3A-AP-22	Design and develop computational artifacts working in team roles using collaborative tools.
Algorithms & Programming	3A-AP-23	Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
Algorithms & Programming	3B-AP-08	Describe how artificial intelligence drives many software and physical systems.
Algorithms & Programming	3B-AP-10	Use and adapt classic algorithms to solve computational problems.
Algorithms & Programming	3B-AP-11	Evaluate algorithms in terms of their efficiency, correctness, and clarity.
Algorithms & Programming	3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules and/or objects.
Algorithms & Programming	3B-AP-16	Demonstrate code reuse by creating programming solutions using libraries and APIs.
Algorithms & Programming	3B-AP-21	Develop and use a series of test cases to verify that a program performs according to its design specifications.
Algorithms & Programming	3B-AP-23	Evaluate key qualities of a program through a process such as a code review.
Computing Systems	3A-CS-02	Compare levels of abstraction and interactions between application software, system software and hardware layers.
Computing Systems	3A-CS-03	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
Data & Analysis	2-DA-07	Represent data using multiple encoding schemes.
Data & Analysis	3A-DA-09	Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.
Data & Analysis	3A-DA-10	Evaluate the tradeoffs in how data elements are organized and where data is stored.
Data & Analysis	3A-DA-11	Create interactive data visualizations using software tools to help others better understand real-world phenomena.
Data & Analysis	3A-DA-12	Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.
Data & Analysis	3B-DA-05	Use data analysis tools and techniques to identify patterns in data representing complex systems.
Data & Analysis	3B-DA-06	Select data collection tools and techniques to generate data sets that support a claim or communicate information.
Impacts of Computing	2-IC-20	Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.
Impacts of Computing	2-IC-23	Describe tradeoffs between allowing information to be public and keeping information private and secure.

Impacts of Computing	3A-IC-24	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
Impacts of Computing	3A-IC-27	Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.
Impacts of Computing	3A-IC-28	Explain the beneficial and harmful effects that intellectual property laws can have on innovation.
Impacts of Computing	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.
Impacts of Computing	3A-IC-30	Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.
Impacts of Computing	3B-IC-25	Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.
Impacts of Computing	3B-IC-26	Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.
Impacts of Computing	3B-IC-27	Predict how computational innovations that have revolutionized aspects of our culture might evolve.
Impacts of Computing	3B-IC-28	Debate laws and regulations that impact the development and use of software.
Networks & the Internet	2-NI-04	Model the role of protocols in transmitting data across networks and the Internet.
Networks & the Internet	3A-NI-04	Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.
Networks & the Internet	3A-NI-05	Give examples to illustrate how sensitive data can be affected by malware and other attacks.
Networks & the Internet	3A-NI-06	Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.
Networks & the Internet	3A-NI-07	Compare various security measures, considering tradeoffs between the usability and security of a computer system.
Networks & the Internet	3B-NI-03	Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).
Networks & the Internet	3B-NI-04	Compare ways software developers protect devices and information from unauthorized access.

RELATED PA ACADEMIC STANDARDS:

- Academic Standards for Business, Computer and Information Technology - Computer and Information Technologies 15.4
- Academic Standards for Science and Technology - Technological Devices 3.7
- Academic Standards for Science and Technology - Science, Technology and Human Endeavors 3.8

PERFORMANCE ASSESSMENTS:

1. Students research and debate dilemmas arising from the digitization of information. (2-IC-20, 3A-IC-24, 3A-IC-28)
2. Students help an imaginary politician design a political stance on a dilemma caused or impacted by the Internet. (3A-IC-24, 3A-IC-28, 3A-IC-30, 3B-IC-26, 3B-IC-28)
3. Students design an app with multiple screens that teaches their classmates about a topic of personal interest. (3A-AP-16, 3A-AP-19, 3A-AP-21, 3A-AP-22, 3A-AP-23)
4. Students build an app that makes a decision or recommendation based on at least two pieces of user input. (2-AP-10, 2-AP-11, 3A-AP-16, 3B-AP-14)
5. Students build an app with any purpose they wish that uses a dataset from App Lab's data library. (3A-AP-14, 3A-AP-16, 3A-AP-22, 3A-AP-23)
6. Students design a library of functions to share with their classmates. (2-AP-14, 3A-AP-18, 3B-AP-14, 3B-AP-16)
7. Students complete AP Create Performance Task, in which they design and implement a program that might solve a problem, enable innovation, explore personal interests, or express creativity. (3A-AP-16, 3A-AP-18, 3A-AP-19, 3A-AP-21, 3A-AP-23, 3B-AP-14)
8. Students choose and analyze a dataset in order to find meaningful patterns and present their findings. (3A-DA-11, 3B-DA-05, 3B-DA-06)
9. Students research and debate computing innovations as they work with a team to make a proposal for a "school of the future". (3A-IC-24, 3A-IC-27, 3B-IC-25)

DESCRIPTION OF COURSE:

AP Computer Science Principles introduces students to the breadth of the field of computer science. In this course, students will learn to design and evaluate solutions and to apply computer science to solve problems through the development of algorithms and programs. They will incorporate abstraction into programs and use data to discover new knowledge. Students will also explain how computing innovations and computing systems, including the Internet, work, explore their potential impacts, and contribute to a computing culture that is collaborative and ethical.

TITLES OF UNITS:

Unit	Week	Title	Overview
1	1-3	Digital Information	Explore how computers store complex information like numbers, text, images and sound and debate the impacts of digitizing information.
2	4-5	The Internet	Learn about how the Internet works and discuss its impacts on politics, culture, and the economy.
3	6-7	Intro to App Design	Design your first app while learning both fundamental programming concepts and collaborative software development processes.
4	8-10	Variables, Conditionals, and Functions	Expand the types of apps you can create by adding the ability to store information, make decisions, and better organize code.
5	11-14	Lists, Loops, and Traversals	Build apps that use large amounts of information and pull in data from the web to create a wider variety of apps.
6	15	Algorithms	Design and analyze algorithms to understand how they work and why some are considered better than others.
7	16-17	Parameters, Return, and Libraries	Learn how to design clean and reusable code that you can share with a single classmate or the entire world.
8	18-21	AP Create Performance Task	Practice and complete the AP Create Performance Task.
9	22-23	Data	Explore and visualize datasets from a wide variety of topics as you hunt for patterns and try to learn more about the world around you.
10	24-26	Cybersecurity and Global Impacts	Research and debate current events at the intersection of data, public policy, law, ethics, and societal impact.

SAMPLE INSTRUCTIONAL STRATEGIES:

1. Programming and Problem-Solving:
    a. Pair programming
    b. Code tracing
    c. Create a plan
    d. Error analysis
    e. Identify a subtask
    f. Look for a patten
    g. Mark the text
    h. Predict and compare
    i. Simplify the problem
    j. Think aloud
    k. Work backward
2. Cooperative Learning:
    a. Discussion group
    b. Jigsaw
    c. Kinesthetic learning
    d. Sharing and responding
    e. Think-pair-share
    f. Unplugged activities
    g. Using manipulatives
3. Making Connections:
    a. Activating prior knowledge
    b. Diagramming
    c. Journaling
    d. Paraphrase
    e. Quickwrite
    f. Vocabulary organizer

MATERIALS:

1. The course requires and assumes a 1:1 computer lab or setup such that each student in the class has access to an internet-connected computer every day in class.
2. Each computer must have a modern web browser installed.
3. All of the course tools and resources (lesson plans, teacher dashboard, videos, student tools, programming environment, etc.) are online and accessible through a modern web browser.
4. It is not required that students have access to internet-connected computers at home to teach this course, but because almost all of the materials are online, it is certainly an advantage.

METHODS OF ASSISTANCE AND ENRICHMENT:

1. Assistance:
    a. Videos to refresh knowledge
    b. Peer programming

    c. Cooperative group work
    d. Formative Assessment using Google Slides with Pear Deck
    e. Class discussion
    f. Conferencing with teacher
    g. Block based vs. text based programming
    h. Brainstorming
    i. Design subtasks
2. Enrichment:
    a. Additional challenges provided in assignments
    b. Open-ended projects which allow students to challenge themselves
    c. Extended projects or reading assignments
    d. Focus on program design

PORTFOLIO DEVELOPMENT:

1. Nine performance assessments listed above, each with an accompanying rubric
2. Online access to all programs developed in class
3. Submission in Canvas of group projects, with peer review

METHODS OF ASSESSMENT:

1. Performance Tasks
2. Computer Programs
3. Classwork
4. Quizzes
5. Self-reflection

INTEGRATED ACTIVITIES:

1. Concepts
    a. Understand abstraction, algorithms and programs
    b. Investigate computer systems and networks
    c. Research and analyze impact of computing
2. Communication
    a. Take a stance on a dilemma impacted by the Internet
    b. Debate dilemmas arising from digitization of information
    c. Present findings to peers
3. Thinking/Problem Solving
    a. Design and evaluate computational solutions for a purpose
    b. Develop programs that incorporate abstractions
    c. Evaluate and test algorithms and programs
    d. Analyze a dataset in order to find meaningful patterns
4. Application of Knowledge

    a. Design an app with multiple screens
    b. Build an app that makes a decision or uses a dataset
    c. Design a library of functions
5. Interpersonal Skills
    a. Collaborate in the development of solutions, e.g. work with a team to make a proposal for a "school of the future"
    b. Utilize “pair programming” to plan, write and troubleshoot programs
    c. Contribute to an inclusive, safe, collaborative, and ethical computing culture