

AP COMPUTER SCIENCE PRINCIPLES STUDY GUIDE

185 High-Yield Concepts

Advanced Placement Course | College Board CED-aligned

-
- All 5 CED Big Ideas: creative development, data, algorithms, networks, impact
 - Exam weights on every section; real AP exam format on the how-to page
 - Algorithms, logic, and code concepts explained without requiring Java
 - Perfect for the AP Computer Science Principles exam in May

TemplateForge | \$19

PDF Workbook | templateforge.com | (c) TemplateForge

How to Use This Guide

This guide condenses the AP Computer Science Principles course and exam description into high-yield concepts. It is not a textbook replacement -- it is the fastest way to review everything the exam covers and to target your weak areas.

The AP Computer Science Principles exam is 2 hours: 70 multiple-choice questions (120 min, 70% of the score), including both single-select and multi-select items. The Create Performance Task (a through-course programming project with written responses) is worth the remaining 30%. The 5 Big Ideas below map to the College Board Course and Exam Description, with exam weights on each section title.

How to use the concepts

- Each concept = term + concise explanation + high-yield exam tip (+ example where useful).
- First pass: read every concept in order. Underline anything unfamiliar.
- Second pass: focus only on the concepts you underlined.
- Two weeks before the exam: rapid-review every tip line (they are the highest-yield).
- Use the Table of Contents to jump straight to your weakest units.

Table of Contents

1	Big Idea 1: Creative Development	36 concepts
2	Big Idea 2: Data	36 concepts
3	Big Idea 3: Algorithms and Programming	45 concepts
4	Big Idea 4: Computing Systems and Networks	34 concepts
5	Big Idea 5: Impact of Computing	34 concepts

Total: 185 high-yield concepts

How to use the TOC: work through the units in order on your first pass, then jump back to your weakest units for review. Each concept includes an Explanation, a high-yield exam tip, and where useful an Example. Mark the concepts you miss with a small dot in the margin and revisit them 48 hours later.

Big Idea 1: Big Idea 1: Creative Development

10-13% of the AP CSP exam. How computing innovations are planned, developed, tested, and improved in teams.

1 Computing Innovation

Explanation: An innovation that includes a computer or program as an integral part, such as apps, websites, robots, and self-driving cars.

AP CSP Tip: *A computing innovation must rely on computing to function -- a toaster is not one, a smart toaster is.*

Example: A fitness tracker is a computing innovation; a plain water bottle is not.

2 Programming as a Creative Process

Explanation: Programming involves creativity: designing solutions, choosing approaches, and refining based on feedback and testing.

AP CSP Tip: *The exam wants you to see programming as design, not just typing code.*

Example: Two programmers can solve the same problem with different, equally valid code.

3 Collaboration in Development

Explanation: Large projects are built by teams using version control, issue tracking, and communication tools to coordinate work.

AP CSP Tip: *Collaboration = dividing work + merging changes; git is the industry standard.*

Example: GitHub lets many developers edit one codebase without overwriting each other.

4 Version Control

Explanation: A system that tracks changes to code over time, allowing multiple developers to work in parallel and roll back mistakes.

AP CSP Tip: *Version control = history + merging + rollback; git is the tool, GitHub the host.*

Example: A developer can revert a broken change to the last working commit.

5 Program Specification

Explanation: A clear description of what a program should do: inputs, outputs, and the behavior expected of it.

AP CSP Tip: *Specifications define requirements BEFORE coding; testing checks code against them.*

Example: A spec for a calculator says: take two numbers and an operator, return the result.

6 Requirements Gathering

Explanation: Determining what users need before building, through interviews, surveys, and observation of existing workflows.

AP CSP Tip: *Requirements = what the user actually needs; skipping them builds the wrong thing.*

Example: Interviewing nurses before building a scheduling app reveals their real workflow.

7 Design Phase

Explanation: Planning the program's structure, user interface, and algorithms before writing code, often with prototypes and diagrams.

AP CSP Tip: *Design precedes code: wireframes, flowcharts, and pseudocode are design artifacts.*

Example: A wireframe of a shopping app's checkout flow is design work.

8 Prototyping

Explanation: Building a quick, working model of a feature or product to test ideas and get feedback early.

AP CSP Tip: *Prototypes = cheap, fast, testable; iterate before committing to a full build.*

Example: A paper prototype of a mobile app tests the layout before any code is written.

9 Iterative Development

Explanation: Building a program in cycles: design, implement, test, evaluate, and repeat, incorporating feedback each round.

AP CSP Tip: *Iteration = build-test-improve loops; agile methods are iterative.*

Example: Each sprint in agile development is one iteration of the cycle.

10 Incremental Development

Explanation: Adding features one small piece at a time, keeping the program working after each addition.

AP CSP Tip: *Incremental = small working steps; contrast with trying to build everything at once.*

Example: A game starts with a moving square, then adds gravity, then enemies.

11 Testing

Explanation: Running a program to find errors (bugs) and verify it meets its specification.

AP CSP Tip: *Testing finds bugs; it cannot prove a program is perfect, only that tested cases work.*

Example: Running a login program with wrong passwords tests error handling.

12 Debugging

Explanation: Finding and fixing errors in a program by tracing its behavior and correcting the code.

AP CSP Tip: *Debugging = locate the cause, not just the symptom; print statements and breakpoints help.*

Example: A program crashes on division; the debugger reveals a zero divisor.

13 Program Documentation

Explanation: Written explanations of what code does and why, helping others (and your future self) understand and maintain it.

AP CSP Tip: *Documentation = comments + manuals; it is part of good programming practice.*

Example: A comment explaining WHY a workaround exists saves future developers hours.

14 Comments in Code

Explanation: Human-readable notes in source code that explain the code's purpose; they are ignored by the computer when running.

AP CSP Tip: *Comments explain intent; they are for people, not the machine.*

Example: `// Calculate total price with tax -- the comment explains the next line.`

15 User Interface (UI)

Explanation: The part of a program the user sees and interacts with: screens, buttons, menus, and controls.

AP CSP Tip: *UI = what the user touches; usability = how easily they do it.*

Example: A mobile app's home screen with icons is its user interface.

16 User Experience (UX)

Explanation: The overall feel of using a product, including ease of use, efficiency, and satisfaction.

AP CSP Tip: *UX = the whole experience; good UI serves good UX.*

Example: One-click checkout is a UX improvement over five-form checkout.

17 Usability

Explanation: How easily and effectively users can accomplish tasks with a product.

AP CSP Tip: *Usability = ease of use; test with real users to find friction.*

Example: A menu with three clear options is more usable than one with thirty.

18 Accessibility

Explanation: Designing products that can be used by people with disabilities: screen readers, color contrast, keyboard navigation.

AP CSP Tip: *Accessibility = design for all; captions, alt text, and contrast are examples.*

Example: Alt text on images lets screen-reader users understand a website.

19 Heuristic Evaluation

Explanation: Reviewing a design against a checklist of usability principles to spot problems without testing.

AP CSP Tip: *Heuristics = rule-of-thumb review; expert evaluation catches obvious issues early.*

Example: Checking that every button has a clear label is a heuristic pass.

20 User Testing

Explanation: Observing real users attempting tasks with a prototype or product to find usability problems.

AP CSP Tip: *User testing = watch real users; their confusion is the data.*

Example: A user testing session reveals that the 'submit' button is invisible.

21 Pair Programming

Explanation: Two programmers working together at one machine: one writes code, the other reviews it in real time, switching roles.

AP CSP Tip: *Pair programming = real-time review; it catches bugs as they are written.*

Example: The 'driver' types while the 'navigator' reviews each line.

22 Feedback Loops

Explanation: Cycles where output feeds back into input, letting a team refine a product continuously.

AP CSP Tip: *Feedback loops = listen, change, re-test; user feedback is the loop's fuel.*

Example: App updates ship monthly based on user reviews -- a feedback loop.

23 Decomposition of a Problem

Explanation: Breaking a large problem into smaller, manageable parts that can be solved and tested separately.

AP CSP Tip: *Decompose = divide and conquer; each part becomes a function or module.*

Example: A weather app splits into data fetch, forecast calculation, and display modules.

24 Modularity

Explanation: Designing a program as separate, interchangeable modules (functions, classes, files) that can be developed and reused independently.

AP CSP Tip: *Modular = independent pieces with clear interfaces; it eases testing and reuse.*

Example: A math library module is reused by many programs.

25 Code Reuse

Explanation: Using existing code -- libraries, functions, templates -- instead of writing everything from scratch.

AP CSP Tip: *Reuse = libraries and functions; it saves time and reduces bugs.*

Example: Using a date-parsing library instead of writing your own parser is code reuse.

26 Libraries and Frameworks

Explanation: Pre-written code collections (libraries) and structures (frameworks) that programmers incorporate into their projects.

AP CSP Tip: *Libraries = tools you call; frameworks = scaffolds you build in.*

Example: React is a framework; a date utility is a library.

27 Intellectual Property

Explanation: Legal rights over creations: copyright (creative works), patents (inventions), and trademarks (brands).

AP CSP Tip: *IP law protects code, music, and designs; violating it is infringement.*

Example: Copying a song without permission violates copyright.

28 Open Source Software

Explanation: Software whose source code is publicly available for anyone to use, modify, and distribute, often under licenses like MIT or GPL.

AP CSP Tip: *Open source = visible + modifiable + free; Linux and Firefox are examples.*

Example: Anyone can read, fork, and improve the Linux kernel.

29 Creative Commons

Explanation: Licenses that let creators grant flexible permissions, such as free use with attribution.

AP CSP Tip: *Creative Commons = share with conditions; CC BY requires credit.*

Example: A photo licensed CC BY can be reused if the photographer is credited.

30 Digital Divide

Explanation: The gap between those with access to technology and the internet and those without, by income, geography, or age.

AP CSP Tip: *The digital divide = unequal access; it affects education and jobs.*

Example: Rural students without broadband struggle with online classes.

31 Malware (Malicious Software)

Explanation: Software designed to harm or exploit: viruses, worms, trojans, ransomware, and spyware.

AP CSP Tip: *Malware = harmful software; keep definitions straight (virus vs worm vs trojan).*

Example: Ransomware encrypts files and demands payment to unlock them.

32 Phishing

Explanation: Fraudulent messages that trick users into revealing passwords, credit cards, or other sensitive information.

AP CSP Tip: *Phishing = fake messages stealing credentials; look for spoofed senders and urgent language.*

Example: An email claiming your bank account is locked and asking for login is phishing.

33 Social Engineering

Explanation: Manipulating people into revealing information or taking unsafe actions, rather than breaking into systems technically.

AP CSP Tip: *Social engineering = attacking the human; phishing is its most common form.*

Example: A caller posing as IT support asking for your password is social engineering.

34 Computer Virus vs Worm

Explanation: A virus needs a host program to spread; a worm spreads on its own across networks without a host.

AP CSP Tip: *Virus = needs a host; worm = self-spreading over networks.*

Example: The Morris worm spread across the early internet without any host program.

35 Ransomware

Explanation: Malware that encrypts a victim's data and demands payment for the decryption key.

AP CSP Tip: *Ransomware = encrypted files + ransom demand; backups are the defense.*

Example: The 2017 WannaCry attack hit hospitals worldwide.

Explanation: A distributed denial-of-service attack floods a server with traffic from many machines to make it unavailable.

AP CSP Tip: *DDoS = overwhelm with traffic; botnets of compromised devices launch them.*

Example: A botnet of thousands of IoT devices can take down a website.

Big Idea 2: Big Idea 2: Data

17-22% of the AP CSP exam. How data is collected, stored, represented, and analyzed, including the digital representation of information.

37 Data

Explanation: Facts and statistics collected for reference or analysis -- the raw material of computing.

AP CSP Tip: *Data = raw facts; information = data with meaning; the exam wants the distinction.*

Example: Temperatures recorded hourly are data; 'today is hotter than yesterday' is information.

38 Binary Representation

Explanation: Computers store everything as binary digits (bits): 0s and 1s. A bit is the smallest unit of data.

AP CSP Tip: *Everything digital -- text, images, sound -- is ultimately bits.*

Example: The letter 'A' is stored as the bit pattern 01000001 in ASCII.

39 Bit and Byte

Explanation: A bit is a single 0 or 1; a byte is 8 bits, enough to represent 256 values.

AP CSP Tip: *8 bits = 1 byte; $2^8 = 256$ possible values per byte.*

Example: A byte can store one ASCII character or a number from 0 to 255.

40 Number Bases: Binary and Hexadecimal

Explanation: Binary is base 2 (0-1), decimal is base 10 (0-9), hexadecimal is base 16 (0-9, A-F). Programs display values in these bases.

AP CSP Tip: *Know binary place values: $1011 = 8+2+1 = 11$ decimal; hex is shorthand for binary.*

Example: The hex color #FF0000 is red: FF = 255 in decimal.

41 Abstraction of Data

Explanation: Using higher-level representations (symbols, names, objects) to hide lower-level complexity (bits, bytes).

AP CSP Tip: *Abstraction = simplify the interface, hide the details.*

Example: The number '42' is an abstraction over the bit pattern that stores it.

42 Data Compression

Explanation: Reducing the size of data by removing redundancy or approximating detail.

AP CSP Tip: *Lossless = perfect reconstruction; lossy = smaller but degraded.*

Example: ZIP files are lossless; JPEG images are lossy.

43 Lossless Compression

Explanation: Compression that allows the original data to be perfectly reconstructed, such as run-length encoding or ZIP.

AP CSP Tip: *Lossless = exact restore; text files must use it.*

Example: Run-length encoding turns AAAAAB into 5A -- reversible exactly.

44 Lossy Compression

Explanation: Compression that discards some detail to save space, such as JPEG and MP3; the original cannot be perfectly restored.

AP CSP Tip: *Lossy = smaller, imperfect; used for images, audio, and video.*

Example: An MP3 drops inaudible frequencies to shrink file size.

45 Data Extraction

Explanation: Pulling relevant data from larger sources, such as scanning sensor feeds or scraping websites for specific values.

AP CSP Tip: *Extraction = select the needed data; cleaning comes next.*

Example: Extracting only temperature readings from a weather dataset.

46 Data Cleaning

Explanation: Fixing or removing errors, duplicates, and inconsistencies in a dataset before analysis.

AP CSP Tip: *Cleaning = garbage in, garbage out prevention; dirty data ruins analysis.*

Example: Removing duplicate customer records before a mailing.

47 Data Filtering

Explanation: Selecting only the data that meets certain criteria, such as rows for a specific year or region.

AP CSP Tip: *Filtering = subset by condition; it is a core spreadsheet and SQL operation.*

Example: Filtering a sales table to show only sales over \$500.

48 Data Visualization

Explanation: Presenting data graphically -- charts, graphs, maps -- to reveal patterns and communicate findings.

AP CSP Tip: *Visualization = patterns you can see; choose the chart that fits the data.*

Example: A line graph of temperatures over a year reveals seasonal patterns.

49 Metadata

Explanation: Data about data: when a file was created, its size, its author, or where a photo was taken.

AP CSP Tip: *Metadata = data describing data; it is often more revealing than the data itself.*

Example: A photo's GPS metadata reveals where it was taken.

50 Open Data

Explanation: Data made freely available for anyone to use and analyze, such as government datasets.

AP CSP Tip: *Open data = public + reusable; it powers research and journalism.*

Example: The US Census publishes open demographic data.

51 Big Data

Explanation: Datasets so large and complex that traditional tools struggle, analyzed with distributed systems.

AP CSP Tip: *Big data = volume, velocity, variety; think petabytes and streams.*

Example: Social media platforms process billions of posts daily.

52 Data Mining

Explanation: Discovering patterns and relationships in large datasets using algorithms, such as market-basket analysis.

AP CSP Tip: *Mining = find patterns you did not know to look for.*

Example: A store finds that customers who buy diapers often buy beer -- a mined pattern.

53 Patterns and Trends

Explanation: Regularities in data -- trends over time, correlations between variables -- that analysis reveals.

AP CSP Tip: *A trend is a general direction; a correlation is a statistical association.*

Example: A decade of rising global temperatures is a trend.

54 Correlation vs Causation

Explanation: Correlation means two variables move together; causation means one causes the other. Correlation does not prove causation.

AP CSP Tip: *The exam loves this: ice cream sales and drownings correlate but neither causes the other.*

Example: A study finds storks and births correlate -- a classic spurious correlation.

55 Sampling

Explanation: Analyzing a subset of a population to draw conclusions about the whole; the sample must be representative.

AP CSP Tip: *Sampling = subset for inference; biased samples give wrong conclusions.*

Example: Polling 1,000 registered voters can predict an election within a few points.

56 Bias in Data

Explanation: Systematic errors in data collection or sampling that skew results, such as non-representative samples.

AP CSP Tip: *Bias = systematic distortion; check who is missing from the data.*

Example: A survey conducted only online misses people without internet access.

57 Surveys and Polls

Explanation: Collecting self-reported data from respondents; question wording and sample selection affect validity.

AP CSP Tip: *Surveys = self-report; leading questions bias answers.*

Example: A poll asking 'do you support wasteful government spending?' skews responses.

58 Sensors and Data Collection

Explanation: Devices that automatically gather data: GPS, cameras, thermometers, heart-rate monitors.

AP CSP Tip: *Sensors = automated data streams; they feed the Internet of Things.*

Example: A smart thermostat collects temperature data every minute.

59 Spreadsheets and Tables

Explanation: Structured data in rows and columns; each row is a record, each column a field.

AP CSP Tip: *Tables = the basic data structure; rows are records, columns are fields.*

Example: A contact list with name, phone, and email columns is a table.

60 SQL and Querying

Explanation: Structured Query Language retrieves and manipulates data from databases (SELECT, WHERE, JOIN).

AP CSP Tip: *SQL = the standard database language; SELECT picks columns, WHERE filters rows.*

Example: SELECT name FROM students WHERE grade > 90 -- a classic query.

61 Database

Explanation: An organized collection of data with tools to store, query, and update it, such as relational databases.

AP CSP Tip: *Databases = organized + queryable; relational databases use tables and keys.*

Example: A library's catalog database stores books, patrons, and loans.

62 Data Privacy

Explanation: Controlling who can access personal data and how it is used; privacy policies and encryption protect it.

AP CSP Tip: *Privacy = control over personal data; the exam ties it to cookies and consent.*

Example: A health app must protect users' medical data from third parties.

63 Cookies

Explanation: Small files websites store on your device to remember you, track sessions, or follow you across sites.

AP CSP Tip: *Cookies = stored identifiers; tracking cookies raise privacy concerns.*

Example: A shopping site uses a cookie to keep your cart between visits.

64 Data Security

Explanation: Protecting data from unauthorized access through encryption, authentication, and access control.

AP CSP Tip: *Security = keep data safe; encryption is the main tool.*

Example: Passwords stored as hashes protect accounts if the server is breached.

65 Encryption

Explanation: Transforming data so only authorized parties can read it, using keys and algorithms.

AP CSP Tip: *Encryption = scramble + key; HTTPS uses it for web traffic.*

Example: The padlock icon in a browser means your connection is encrypted.

66 Symmetric vs Asymmetric Encryption

Explanation: Symmetric encryption uses one key for both encrypting and decrypting; asymmetric (public-key) uses a public key to encrypt and a private key to decrypt.

AP CSP Tip: *Asymmetric = public/private key pair; it solves secure key exchange.*

Example: HTTPS uses asymmetric encryption to exchange a session key, then symmetric for speed.

67 Moore's Law

Explanation: The observation that computing power roughly doubles every two years as transistors shrink, driving data processing growth.

AP CSP Tip: *Moore's law = exponential hardware growth; it is slowing as limits approach.*

Example: A modern phone has more computing power than 1990s supercomputers.

68 Lossless vs Lossy File Formats

Explanation: Formats trade size for fidelity: PNG (lossless) vs JPEG (lossy), WAV vs MP3.

AP CSP Tip: *Match formats to needs: text and archives lossless; photos and music can be lossy.*

Example: A screenshot should be PNG; a vacation photo can be JPEG.

69 Data Scalability

Explanation: Systems must handle growing data volumes without breaking; distributed storage scales horizontally.

AP CSP Tip: *Scalability = handle more data; sharding and cloud storage scale it.*

Example: A social network's data warehouse grows by petabytes yearly.

70 Data Ethics

Explanation: The responsible use of data: consent, transparency, fairness, and avoiding harm.

AP CSP Tip: *Data ethics = responsible use; bias and privacy are the core issues.*

Example: Using customer data for purposes they did not consent to is an ethics violation.

71 Artificial Intelligence and Data

Explanation: AI systems learn patterns from large datasets; their quality depends on the data they train on.

AP CSP Tip: *AI = pattern learning from data; biased training data makes biased AI.*

Example: A hiring AI trained on past hires may replicate past discrimination.

Explanation: Everyday devices connected to the internet, collecting and sharing data: smart home devices, wearables, sensors.

AP CSP Tip: *IoT = connected everyday objects; they generate continuous data streams.*

Example: A smart doorbell sends video and motion data to the cloud.

Big Idea 3: Big Idea 3: Algorithms and Programming

30-35% of the AP CSP exam. The largest unit: algorithms, program logic, control structures, and code correctness.

73 Algorithm

Explanation: A finite sequence of steps to solve a problem, independent of any particular programming language.

AP CSP Tip: *Algorithms = step-by-step recipes; they can be expressed as flowcharts or pseudocode.*

Example: A recipe for making coffee is an algorithm.

74 Pseudocode

Explanation: A plain-language description of an algorithm that is precise enough to be turned into code.

AP CSP Tip: *Pseudocode = the algorithm before the syntax; AP CSP questions use it.*

Example: IF score >= 90 THEN print 'A' -- pseudocode for grading.

75 Flowchart

Explanation: A diagram of an algorithm using shapes: ovals (start/end), rectangles (process), diamonds (decisions).

AP CSP Tip: *Flowcharts = visual algorithms; diamonds are decisions.*

Example: A login flowchart shows a diamond for 'valid password?'.

76 Sequence

Explanation: The order in which instructions are executed, one after another, from top to bottom.

AP CSP Tip: *Sequence = the default flow; control structures change it.*

Example: x = 5 then y = x + 1 executes in sequence.

77 Selection (Conditionals)

Explanation: Choosing between paths based on a condition, using IF, ELSE IF, and ELSE.

AP CSP Tip: *Selection = the branch; conditions evaluate to true or false.*

Example: IF age >= 18 THEN print 'adult' ELSE print 'minor'.

78 Iteration (Loops)

Explanation: Repeating a block of code: FOR loops (known count) and WHILE loops (condition-based).

AP CSP Tip: *Iteration = repetition; FOR counts, WHILE repeats while true.*

Example: FOR i = 1 TO 10: print i -- loops ten times.

79 FOR Loop

Explanation: A loop that repeats a fixed number of times with a counter variable.

AP CSP Tip: *FOR loops = definite iteration; the counter is automatic.*

Example: FOR i = 1 TO 5: total = total + i.

80 WHILE Loop

Explanation: A loop that repeats as long as a condition is true; it may run zero or infinite times.

AP CSP Tip: *WHILE = condition-controlled; ensure the condition eventually becomes false.*

Example: WHILE balance > 0: withdraw(10) -- runs until balance hits zero.

81 Nested Loops

Explanation: A loop inside another loop; the inner loop completes fully for each outer iteration.

AP CSP Tip: *Nested loops = multiplication of repetitions; $O(n^2)$ behavior.*

Example: For each row, for each column: print "*" -- draws a grid.

82 Boolean Expression

Explanation: An expression that evaluates to true or false, using comparisons (`==`, `<`, `>`) and logic operators (`AND`, `OR`, `NOT`).

AP CSP Tip: *Booleans = true/false; AND requires both true, OR requires at least one.*

Example: `(age >= 18) AND (has_id)` -- true only if both hold.

83 AND / OR / NOT Logic

Explanation: AND is true only when both operands are true; OR is true when at least one is; NOT flips the value.

AP CSP Tip: *Truth tables: AND = both; OR = either; NOT = inverse.*

Example: NOT (raining) is true when it is not raining.

84 Variables

Explanation: Named storage for values that can change during program execution.

AP CSP Tip: *Variables = labeled boxes; assign with `=`, use the value later.*

Example: `score = 0` then `score = score + 10` changes the stored value.

85 Constants

Explanation: Named values that do not change during a program, such as `PI` or `MAX_SPEED`.

AP CSP Tip: *Constants = fixed values; naming them clarifies code.*

Example: `const TAX_RATE = 0.08` -- used throughout a price calculator.

86 Data Types

Explanation: Kinds of values: integers, floats (decimals), strings (text), and booleans.

AP CSP Tip: *Match types to data: numbers, text, and true/false are the basics.*

Example: 'hello' is a string; 3.14 is a float; 7 is an integer.

87 Integer vs Float

Explanation: Integers are whole numbers; floats have decimal parts. Division can produce floats from integers.

AP CSP Tip: *Know when a result is an integer vs a float -- `7/2 = 3.5` (float).*

Example: In many languages, `7/2` is 3.5 but integer division `7//2` is 3.

88 String

Explanation: A sequence of characters, manipulated with concatenation, length, and substring operations.

AP CSP Tip: *Strings = text; `+` concatenates, `len()` measures.*

Example: `'ab' + 'cd' = 'abcd'`; `len('abcd') = 4`.

89 Concatenation

Explanation: Joining strings end to end with the `+` operator (or language equivalent).

AP CSP Tip: *Concatenation = glue strings; numbers may need conversion first.*

Example: `'Hello' + 'world' = 'Hello world'`.

90 Lists (Arrays)

Explanation: Ordered collections of values accessible by index, starting at 0 in most languages.

AP CSP Tip: *Lists = indexed collections; remember zero-based indexing.*

Example: `items = [10, 20, 30]; items[1]` is 20.

91 List Operations

Explanation: Adding, removing, and accessing elements: `append`, `insert`, `remove`, and index lookup.

AP CSP Tip: *Know the four basics: add, remove, access by index, modify in place.*

Example: `append(items, 40)` grows the list to `[10, 20, 30, 40]`.

92 Indexing

Explanation: Accessing a specific element by its position; most languages start at index 0.

AP CSP Tip: *Index 0 = first element; index 1 = second; off-by-one errors are classic bugs.*

Example: In `[5, 6, 7]`, the value 5 is at index 0, 7 is at index 2.

93 Dictionary (Map)

Explanation: A collection of key-value pairs, retrieving values by unique keys.

AP CSP Tip: *Dictionaries = key -> value lookup; keys must be unique.*

Example: `grades = {'Ada': 95, 'Bob': 87}; grades['Ada']` is 95.

94 Function

Explanation: A named block of reusable code that takes inputs (parameters) and returns an output.

AP CSP Tip: *Functions = reusable units; parameters in, return value out.*

Example: `def square(x): return x * x` -- `square(5)` = 25.

95 Parameters and Arguments

Explanation: Parameters are the named inputs in a function definition; arguments are the actual values passed when called.

AP CSP Tip: *Definition has parameters; the call passes arguments.*

Example: In `square(x)`, `x` is the parameter; `square(5)` passes 5 as the argument.

96 Return Value

Explanation: The value a function sends back to the caller, used in the calling expression.

AP CSP Tip: *return = the function's output; without it, the call yields nothing useful.*

Example: `def add(a, b): return a + b` -- `add(2, 3)` evaluates to 5.

97 Modular Program Structure

Explanation: Organizing a program into small, single-purpose functions and modules for clarity and reuse.

AP CSP Tip: *Modular structure = small functions, one job each.*

Example: A banking app has separate functions for deposit, withdraw, and balance.

98 Program Correctness

Explanation: Whether a program produces the expected output for all valid inputs, verified by testing and reasoning.

AP CSP Tip: *Correctness = matches the spec; test edge cases, not just typical ones.*

Example: Testing a division program with a zero divisor checks correctness at the edge.

99 Test Cases

Explanation: Specific inputs with expected outputs used to verify a program behaves correctly.

AP CSP Tip: *Test cases = input-output pairs; include edge and boundary cases.*

Example: For a max function, test negative numbers, ties, and the empty list.

100 Edge Cases

Explanation: Unusual or extreme inputs: empty lists, zero values, very large numbers, boundary conditions.

AP CSP Tip: *Edge cases = where bugs hide; always test them.*

Example: A search function should handle an empty search box gracefully.

101 Bugs and Errors

Explanation: Defects in code that cause wrong behavior; syntax errors break compilation, logic errors produce wrong results.

AP CSP Tip: *Syntax = the language rules; logic = the algorithm's correctness.*

Example: A missing parenthesis is a syntax error; a wrong formula is a logic error.

102 Off-by-One Errors

Explanation: A classic logic bug where a loop runs one time too many or too few, often from index confusion.

AP CSP Tip: *Off-by-one = boundary mistake; check loop start and end conditions.*

Example: Looping i from 1 to $\text{len}(\text{list})$ instead of 0 to $\text{len}(\text{list})-1$ skips or repeats an element.

103 Linear Search

Explanation: Scanning a list from start to finish until the target is found; $O(n)$ time.

AP CSP Tip: *Linear search = check each item; fine for small or unsorted lists.*

Example: Looking for 'Ada' by checking each name in order.

104 Binary Search

Explanation: Repeatedly halving a sorted list to find a target; $O(\log n)$ time, much faster than linear search.

AP CSP Tip: *Binary search = halving; it requires a SORTED list.*

Example: Guessing a number between 1 and 100 by halving takes at most 7 guesses.

105 Sorting Algorithms

Explanation: Arranging data in order: selection sort and insertion sort are simple $O(n^2)$; merge sort and quicksort are faster $O(n \log n)$.

AP CSP Tip: *Know that sorting enables faster search and that fast sorts are $O(n \log n)$.*

Example: Sorting a contact list alphabetically before searching it.

106 Selection Sort

Explanation: Repeatedly finding the smallest element and swapping it into place; $O(n^2)$ time.

AP CSP Tip: *Selection sort = find min, swap; simple but slow on big lists.*

Example: Sorting 5 numbers takes up to 5 find-and-swap passes.

107 Merge Sort

Explanation: A divide-and-conquer sort that splits the list in half, sorts each half, and merges them; $O(n \log n)$.

AP CSP Tip: *Merge sort = split, sort, merge; it is reliably $O(n \log n)$.*

Example: Splitting a 8-item list into singles then merging pairs back up.

108 Linear vs Binary Search Efficiency

Explanation: Linear search is $O(n)$; binary search is $O(\log n)$ but needs sorted data. For large sorted lists, binary wins.

AP CSP Tip: *Efficiency = how time grows with input size; the exam asks for comparisons.*

Example: Binary search on a billion items takes ~30 steps; linear takes up to a billion.

109 Heuristic Algorithm

Explanation: A practical, approximate approach that finds a good-enough solution quickly when an exact one is impractical.

AP CSP Tip: *Heuristics = good enough, fast; used in search and routing.*

Example: A GPS approximates the fastest route without checking every path.

110 Undecidable Problem

Explanation: A problem with no algorithm that can solve it for all inputs, such as the halting problem.

AP CSP Tip: *Undecidable = no general algorithm exists; the halting problem is the example.*

Example: No program can decide, for every program, whether it will halt.

111 Reasoning About Programs

Explanation: Analyzing code by tracing execution step by step or by logical argument, to predict output or find flaws.

AP CSP Tip: *Trace the code: simulate each line with sample values.*

Example: Tracing $x = 0$; $x = x + 1$; $x = x * 2$ gives $x = 2$.

112 Tracing Code

Explanation: Manually stepping through a program's execution, recording variable values at each line.

AP CSP Tip: *Tracing = the debugger in your head; keep a table of variable values.*

Example: Trace a loop by writing down the counter and total after each pass.

113 Control Flow

Explanation: The order of statement execution, directed by sequence, selection, and iteration.

AP CSP Tip: *Control flow = sequence + branches + loops; the skeleton of any program.*

Example: A grading program flows: read score, branch on ranges, print letter.

114 Data Abstraction in Programming

Explanation: Representing complex data behind simple interfaces, such as a list hiding its internal storage.

AP CSP Tip: *Abstraction = simplicity at the interface, complexity hidden.*

Example: A contacts list abstracts away how addresses are stored in memory.

115 Procedural Abstraction

Explanation: Hiding how a task is done inside a function, so callers only need its name and inputs.

AP CSP Tip: *Procedural abstraction = functions as black boxes.*

Example: A 'sort' function abstracts away the sorting steps from its caller.

116 Collaboration and Code Quality

Explanation: Readable, documented, modular code is easier for teams to maintain and extend.

AP CSP Tip: *Quality code = readable + tested + documented; maintainability matters.*

Example: Clear function names and comments make a codebase team-friendly.

Explanation: Unit tests (one function), integration tests (parts together), and user acceptance tests (meets the need).

AP CSP Tip: *Test at all levels: units, integration, acceptance.*

Example: A unit test checks one function; an integration test checks the full checkout flow.

Big Idea 4: Big Idea 4: Computing Systems and Networks

11-15% of the AP CSP exam. How computers and networks work and communicate.

118 Computer System

Explanation: A combination of hardware (physical parts) and software (programs) that processes data.

AP CSP Tip: *Hardware = physical; software = instructions; both are needed.*

Example: A laptop's CPU is hardware; its operating system is software.

119 Central Processing Unit (CPU)

Explanation: The computer's processor, executing instructions: fetch, decode, execute.

AP CSP Tip: *The CPU = the brain; clock speed and cores affect performance.*

Example: A CPU fetches an instruction, decodes it, and executes it billions of times per second.

120 Random Access Memory (RAM)

Explanation: Volatile memory holding data and programs while the computer runs; it is cleared on shutdown.

AP CSP Tip: *RAM = fast, temporary, volatile; more RAM = smoother multitasking.*

Example: Closing an app frees its RAM for other programs.

121 Storage (Hard Drive / SSD)

Explanation: Persistent memory that keeps data when the power is off: hard drives (magnetic) and SSDs (flash).

AP CSP Tip: *Storage = long-term; RAM = short-term. Do not confuse them.*

Example: Your photos survive a reboot because they are on storage, not RAM.

122 Input and Output Devices

Explanation: Input devices (keyboard, mouse, microphone) send data in; output devices (screen, speaker, printer) present results.

AP CSP Tip: *Input = in, output = out; identify each in examples.*

Example: A touchscreen is both an input (touch) and output (display) device.

123 Operating System (OS)

Explanation: Software managing hardware and running applications: Windows, macOS, Linux, iOS, Android.

AP CSP Tip: *The OS = the manager; it handles files, memory, and processes.*

Example: When you open a file, the OS coordinates the disk, RAM, and the app.

124 Software

Explanation: Programs and instructions that tell hardware what to do: system software and application software.

AP CSP Tip: *Software = instructions; applications are software you use directly.*

Example: A word processor is application software; the OS is system software.

125 Application Software

Explanation: Programs for specific user tasks: browsers, editors, games, and productivity tools.

AP CSP Tip: *Apps = task-focused software; they run on top of the OS.*

Example: Chrome, Excel, and Spotify are application software.

126 Programs and Processes

Explanation: A program is stored code; a process is a program in execution, managed by the OS.

AP CSP Tip: *Program = file; process = running instance.*

Example: Opening the same game twice creates two processes from one program.

127 Von Neumann Architecture

Explanation: The classic computer design: CPU, memory, and I/O connected by a bus, storing programs in memory.

AP CSP Tip: *Von Neumann = stored-program computer; most computers follow it.*

Example: Your laptop fetches program instructions from memory just like data.

128 Abstraction in Computing Systems

Explanation: Layers hide complexity: applications use the OS, which uses the CPU, without knowing hardware details.

AP CSP Tip: *Layers = abstraction; each level hides the level below.*

Example: A browser sends 'open this page' without managing network packets directly.

129 Latency

Explanation: The time delay before data transfer begins; high latency makes connections feel slow.

AP CSP Tip: *Latency = delay; bandwidth = capacity; a fast highway (bandwidth) with traffic lights (latency) still feels slow.*

Example: Gamers dislike high latency because actions lag.

130 Bandwidth

Explanation: The amount of data that can be transmitted per unit of time, like the width of a pipe.

AP CSP Tip: *Bandwidth = data rate; measured in bits per second.*

Example: Streaming 4K video needs higher bandwidth than streaming audio.

131 Computer Network

Explanation: Connected devices that share data and resources, from home Wi-Fi to the global internet.

AP CSP Tip: *Networks = connected devices; the internet is the network of networks.*

Example: Your phone and laptop sharing a printer are on a home network.

132 The Internet

Explanation: The global system of interconnected networks using standard protocols (TCP/IP) to exchange data.

AP CSP Tip: *The internet = networks + standards; TCP/IP is its foundation.*

Example: Sending an email crosses many networks using IP addressing.

133 IP Address

Explanation: A unique numeric identifier for a device on a network, such as 192.168.1.1 or an IPv6 address.

AP CSP Tip: *IP addresses = network locations; devices need one to communicate.*

Example: Your router assigns your devices local IP addresses.

134 Domain Name System (DNS)

Explanation: The 'phone book' translating human-readable domain names (example.com) into IP addresses.

AP CSP Tip: *DNS = name -> IP lookup; without it you would type numbers, not names.*

Example: Typing repurposeforge.com triggers a DNS lookup to find its server.

135 Packet

Explanation: A small unit of data transmitted over a network; messages are split into packets and reassembled at the destination.

AP CSP Tip: *Packets = data chunks; they may take different routes and arrive out of order.*

Example: A photo upload splits into thousands of packets across the internet.

136 Router

Explanation: A device forwarding packets between networks, choosing paths to their destinations.

AP CSP Tip: *Routers = the internet's traffic cops, directing packets hop by hop.*

Example: A home router connects your Wi-Fi devices to your internet service provider.

137 Protocol

Explanation: A set of rules for communication, such as HTTP, HTTPS, TCP, and IP.

AP CSP Tip: *Protocols = agreed rules; both sides must follow them to communicate.*

Example: HTTP defines how browsers request and receive web pages.

138 TCP/IP

Explanation: The internet's core protocols: IP addresses packets; TCP ensures reliable, ordered delivery.

AP CSP Tip: *IP = addressing, TCP = reliable delivery; the internet's foundation.*

Example: TCP retransmits lost packets so downloads arrive complete.

139 HTTP and HTTPS

Explanation: HTTP is the web's request-response protocol; HTTPS adds encryption for security.

AP CSP Tip: *HTTPS = HTTP + encryption (TLS); always look for the padlock.*

Example: Banking sites force HTTPS so data cannot be read in transit.

140 URL

Explanation: A Uniform Resource Locator identifying a resource: protocol + domain + path.

AP CSP Tip: *URLs = web addresses; they name resources on the internet.*

Example: <https://repurposeforge.com/products> is a URL: HTTPS, domain, path.

141 Client-Server Model

Explanation: Clients request services; servers provide them, such as a browser (client) fetching from a web server.

AP CSP Tip: *Client asks, server answers; the dominant web model.*

Example: Your phone (client) requests a map from a server, which sends tiles.

142 Peer-to-Peer (P2P)

Explanation: Devices communicate directly without a central server, sharing resources equally.

AP CSP Tip: *P2P = no central hub; used by file-sharing and some messaging apps.*

Example: BitTorrent downloads pieces of a file from many peers at once.

143 Cloud Computing

Explanation: Delivering computing services (storage, processing, apps) over the internet from remote data centers.

AP CSP Tip: *Cloud = remote resources on demand; it shifts cost from ownership to rental.*

Example: Google Drive stores files in the cloud, not on your device.

144 Scalability of the Internet

Explanation: The internet scales by distributing work across many devices and networks rather than one central machine.

AP CSP Tip: *The internet = distributed by design; that is why it scales and survives failures.*

Example: If one server fails, other servers and routes keep the internet working.

145 Fault Tolerance

Explanation: A system's ability to keep working when parts fail, through redundancy and distributed design.

AP CSP Tip: *Fault tolerance = survive failures; redundancy is the mechanism.*

Example: Redundant servers mean a single hardware failure does not take down a service.

146 Redundancy

Explanation: Duplicating components or paths so a failure does not break the system.

AP CSP Tip: *Redundancy = backup paths and copies; the internet is built on it.*

Example: Email routes around a broken cable through redundant paths.

147 The Internet of Things (Networks)

Explanation: Everyday objects with network connectivity: smart bulbs, thermostats, cameras, and appliances.

AP CSP Tip: *IoT = connected devices; they add convenience and security risks.*

Example: A smart lock connects to your phone over the home network.

148 Wireless Networks (Wi-Fi, Bluetooth)

Explanation: Radio-based networking: Wi-Fi for local area connections, Bluetooth for short-range device links.

AP CSP Tip: *Wi-Fi = local wireless; Bluetooth = short-range personal links.*

Example: Wireless headphones pair via Bluetooth; the laptop uses Wi-Fi.

149 Mobile Networks (4G/5G)

Explanation: Cellular networks covering wide areas, letting phones connect anywhere with a signal.

AP CSP Tip: *Cellular = wide-area wireless; each generation adds speed and capacity.*

Example: Streaming video on a phone uses the 5G cellular network.

150 Security in Networks

Explanation: Network defenses include encryption, firewalls, and authentication to protect data in transit.

AP CSP Tip: *Network security = protect data as it travels; encryption is the key tool.*

Example: VPNs encrypt traffic to protect it on public Wi-Fi.

151 Firewall

Explanation: A system filtering incoming and outgoing network traffic, blocking unauthorized access.

AP CSP Tip: *Firewalls = network filters; they block unwanted connections.*

Example: A home router's firewall blocks unsolicited incoming connections.

Big Idea 5: Big Idea 5: Impact of Computing

21-26% of the AP CSP exam. Computing's effects on society, the economy, and culture, plus responsible use.

152 Impact of Computing

Explanation: Computing transforms how people work, communicate, learn, and live -- with both benefits and harms.

AP CSP Tip: *Impacts = benefits AND harms; the exam wants both sides.*

Example: Social media connects people but can spread misinformation.

153 Automation

Explanation: Using computers and machines to perform tasks without direct human control, replacing or augmenting human labor.

AP CSP Tip: *Automation = machines do the work; it raises productivity and displaces jobs.*

Example: Automated checkout lanes replace some cashier roles.

154 Artificial Intelligence (AI)

Explanation: Computer systems that perform tasks requiring human-like intelligence: learning, reasoning, perception.

AP CSP Tip: *AI = learning and reasoning systems; they depend on data and algorithms.*

Example: Voice assistants use AI to understand and answer questions.

155 Machine Learning

Explanation: A subset of AI where systems learn patterns from data rather than following explicit rules.

AP CSP Tip: *ML = learn from data; more data often means better models.*

Example: Spam filters learn to recognize spam from labeled examples.

156 Bias in Computing

Explanation: Systems can reflect or amplify human bias through biased data, design, or use.

AP CSP Tip: *Bias = unfair outcomes; check training data and design choices.*

Example: A facial recognition system trained mostly on light skin has higher error rates for darker skin.

157 Accessibility of Computing

Explanation: Designing technology usable by people with disabilities: screen readers, captions, voice control.

AP CSP Tip: *Accessibility = design for all abilities; it is both ethical and good business.*

Example: Closed captions make videos accessible to deaf viewers.

158 The Digital Divide Revisited

Explanation: Unequal access to technology and digital skills by income, geography, age, and education.

AP CSP Tip: *The divide = who is left out of the digital economy.*

Example: Students without home internet fall behind in online learning.

159 Globalization and Computing

Explanation: Computing enables global communication, trade, and collaboration, connecting economies and cultures.

AP CSP Tip: *Computing = the connective tissue of globalization.*

Example: A team in four time zones collaborates through video calls and shared docs.

160 E-Commerce

Explanation: Buying and selling goods and services online, reshaping retail and creating global markets.

AP CSP Tip: *E-commerce = online trade; it lowers barriers for small sellers.*

Example: A small shop can sell worldwide through an online storefront.

161 Gig Economy

Explanation: Work mediated by digital platforms: ride-sharing, delivery, freelance marketplaces.

AP CSP Tip: *The gig economy = platform-mediated, flexible, often precarious work.*

Example: Drivers and delivery workers connect to customers through apps.

162 Remote Work

Explanation: Working from anywhere with internet access, enabled by collaboration tools.

AP CSP Tip: *Remote work = work without a fixed office; it reshapes cities and commuting.*

Example: A software engineer works from home with cloud tools and video calls.

163 Social Media

Explanation: Platforms for sharing content and connecting, with effects on communication, politics, and mental health.

AP CSP Tip: *Social media = connection + risks (misinformation, echo chambers, harassment).*

Example: News spreads on social platforms faster than traditional media.

164 Echo Chambers and Filter Bubbles

Explanation: Algorithms that feed users content matching their views, isolating them from opposing perspectives.

AP CSP Tip: *Filter bubbles = algorithm-driven isolation; they polarize discourse.*

Example: A user who watches one political view sees more of the same.

165 Misinformation and Disinformation

Explanation: Misinformation is false information shared without intent to deceive; disinformation is deliberately false.

AP CSP Tip: *Misinformation = accidental; disinformation = deliberate. Both are harmful.*

Example: Sharing an outdated warning is misinformation; a forged video is disinformation.

166 Digital Footprint

Explanation: The trail of data you leave online: posts, searches, purchases, and location data.

AP CSP Tip: *Digital footprints are permanent and searchable; think before you post.*

Example: A college admissions officer can find a candidate's old public posts.

167 Data Collection and Consent

Explanation: Companies collect user data; consent and transparency are the ethical questions.

AP CSP Tip: *Consent = informed agreement; read what you agree to.*

Example: An app asking to access your contacts should explain why.

168 Privacy and Surveillance

Explanation: The tension between personal privacy and government or corporate surveillance.

AP CSP Tip: *Privacy = control over personal information; surveillance = observation.*

Example: Location tracking by apps raises privacy concerns.

169 Intellectual Property Revisited

Explanation: Computing makes copying easy, straining copyright and patents; licensing (open source, Creative Commons) adapts.

AP CSP Tip: *Digital copying = easy; licensing defines what is legal.*

Example: Streaming platforms license music rather than selling copies.

170 Piracy

Explanation: Unauthorized copying or distribution of copyrighted digital works.

AP CSP Tip: *Piracy = unauthorized copying; it is illegal and common digitally.*

Example: Downloading a movie from an unauthorized site is piracy.

171 Censorship

Explanation: Government or platform control of information and speech, enabled or resisted by technology.

AP CSP Tip: *Censorship = restricting access; encryption and VPNs can bypass it.*

Example: Some governments block access to certain websites.

172 Freedom of Speech Online

Explanation: The internet enables broad speech but platforms moderate content, creating policy tensions.

AP CSP Tip: *Speech vs moderation = a live policy debate.*

Example: A platform removing hate speech while allowing political criticism balances values.

173 Cybersecurity for Users

Explanation: Individuals protect themselves with strong passwords, updates, and awareness of phishing.

AP CSP Tip: *User security = passwords + updates + skepticism.*

Example: Using a password manager and two-factor authentication protects accounts.

174 Two-Factor Authentication (2FA)

Explanation: Requiring a second proof of identity (code, biometric) beyond a password.

AP CSP Tip: *2FA = password + second factor; it blocks most account takeovers.*

Example: Entering a code from your phone after your password is 2FA.

175 Strong Passwords

Explanation: Long, unique, random passwords per account; password managers help.

AP CSP Tip: *Strong = long + unique; reuse is the enemy.*

Example: 'correct horse battery staple' beats short, common passwords.

176 Symmetric vs Public Key Encryption (Review)

Explanation: Encryption protects data; public-key methods enable secure communication without sharing secrets.

AP CSP Tip: *Encryption = the backbone of online trust; HTTPS uses it everywhere.*

Example: Your browser verifies a site's certificate using public keys.

177 Digital Citizenship

Explanation: Responsible online behavior: respect, honesty, and legal compliance.

AP CSP Tip: *Digital citizenship = be respectful and responsible online.*

Example: Citing sources and not plagiarizing is digital citizenship.

178 Cybersecurity Careers and Skills

Explanation: Computing offers high-demand careers: software, security, data, and network roles.

AP CSP Tip: *Computing skills = careers; cybersecurity is one of the fastest-growing fields.*

Example: Security analysts monitor networks for threats.

179 Economic Impact of Computing

Explanation: Computing creates new industries and jobs while automating others, shifting the economy.

AP CSP Tip: *Computing = creative destruction: new jobs replace old ones.*

Example: App stores created an entire developer economy.

180 Environmental Impact of Computing

Explanation: Computing consumes energy and resources; data centers and device production have environmental costs.

AP CSP Tip: *Green computing = efficiency and sustainability in tech.*

Example: Cloud providers are moving to renewable energy for data centers.

181 E-Waste

Explanation: Discarded electronics, often shipped to developing countries for dismantling, with health and environmental costs.

AP CSP Tip: *E-waste = the hidden cost of constant upgrades; recycling helps.*

Example: Old phones and laptops often end up in e-waste streams abroad.

182 Computing and Education

Explanation: Technology transforms learning: online courses, personalized tools, and global access to knowledge.

AP CSP Tip: *EdTech = access + personalization; the digital divide limits it.*

Example: Massive open online courses reach millions of learners.

183 Healthcare Computing

Explanation: AI diagnosis, electronic records, telehealth, and wearable monitoring improve care.

AP CSP Tip: *Health tech = better diagnosis and access; privacy is the concern.*

Example: Wearable heart monitors alert doctors to problems remotely.

184 Government and Computing

Explanation: Governments use data and algorithms for services, security, and elections -- with accountability questions.

AP CSP Tip: *GovTech = efficiency vs surveillance; transparency matters.*

Example: Online tax filing simplifies services; automated welfare decisions raise fairness questions.

185 Future of Computing

Explanation: Quantum computing, AI, and ubiquitous connectivity promise transformations -- and new risks.

AP CSP Tip: *Future tech = potential + uncertainty; assess both.*

Example: Quantum computers may crack current encryption, driving new security research.

What's Next?

You have now reviewed all five Big Ideas of the AP Computer Science Principles course. Go back to the Table of Contents, find the sections you marked, and reread just those tip lines. Two weeks out, do a rapid pass over the entire guide; the day before, review only the tip lines. For the multiple-choice exam, eliminate wrong answers and watch for absolutes; for the Create Performance Task, document your development process, acknowledge collaboration, and explain how your program handles data and abstraction.

Pair it with the Study Mastery Cheat Sheet (\$19) -- how to actually retain all of it

Also available: AP Computer Science A, AP Biology, AP Chemistry, and more study guides

templateforge.com