Computer Science Principles
PLTW Framework
Course Level

PLTW Framework - Overview

PLTW Frameworks are representations of the knowledge, skills, and understandings that empower students to thrive in an evolving world. The PLTW Frameworks define the scope of learning and instruction within the PLTW curricula. The framework structure is organized by four levels of understanding that build upon each other: Knowledge and Skills, Objectives, Domains, and Competencies.

The most fundamental level of learning is defined by course Knowledge and Skills statements. Each Knowledge and Skills statement reflects specifically what students will know and be able to do after they've had the opportunity to learn the course content. Students apply Knowledge and Skills to achieve learning Objectives, which are skills that directly relate to the workplace or applied academic settings. Objectives are organized by higher-level Domains.

Domains are areas of in-demand expertise that an employer in a specific field may seek; they are key understandings and long-term takeaways that go beyond factual knowledge into broader, conceptual comprehension.

At the highest level, Competencies are general characterizations of the transportable skills that benefit students in various professional and academic pursuits. As a whole, the PLTW Frameworks illustrate the deep and relevant learning opportunities students experience from PLTW courses and demonstrate how the courses prepare students for life, not just the next grade level.

To thrive in an evolving world, students need skills that will benefit them regardless of the career path they choose. PLTW Frameworks are organized to showcase alignment to in-demand, transportable skills. This alignment ensures that students learn skills that are increasingly important in the rapidly advancing, innovative workplace.

Competencies (C), Domains (D), Objectives (O), Knowledge and Skills (KS)

C1 Problem Solving and Process Thinking

Strategic and systematic design and inquiry processes guide the development of an effective solution to the problem.

D1 Creativity

Computing is a creative activity. Creativity and computing are prominent forces in innovation; the innovations enabled by computing have had and will continue to have far-reaching impact.

O1.1 Apply a creative development process when creating computational artifacts. LO 1.1.1 [P2]

KS1.1.1 Understand that a creative process in the development of a computational artifact can include, but is not limited to, employing nontraditional, nonprescribed techniques; the use of novel combinations of artifacts, tools, and techniques; and the exploration of personal curiosities.

KS1.1.2 Translate ideas into tangible form by creating computational artifacts and employing an iterative and exploratory process. EK 1.1.1B

O1.2 (1V) Create a computational artifact for creative expression. LO 1.2.1 [P2]

KS1.2.1 Identify a computational artifact as something created by a human using a computer and differentiate between a program, an image, an audio, a video, a presentation, or a web page file. EK 1.2.1A

KS1.2.2 Understand that a creatively developed computational artifact can be created by using nontraditional, nonprescribed computing techniques. EK 1.2.1D

KS1.2.3 Understand that creative expressions in a computational artifact can reflect personal expressions of ideas or interests. EK 1.2.1E

O1.3 Create a new computational artifact by combining or modifying existing artifacts. [P2] LO 1.2.3

KS1.3.1 Understand that creating computational artifacts can be done by combining and modifying existing artifacts or by creating new artifacts. EK 1.2.3A

KS1.3.2 Understand that computation facilitates the creation and modification of computational artifacts with enhanced detail and precision. EK 1.2.3B

KS1.3.3 Understand that combining or modifying existing artifacts can show personal expression of ideas. EK 1.2.3C

O1.4 Use computing tools and techniques for creative expression. [P2] LO 1.3.1

KS1.4.1 Understand that creating digital effects, images, audio, video, and animations has transformed industries. EK 1.3.1A

KS1.4.3 Understand that digital images can be created by generating pixel patterns, manipulating existing digital images, or combining images. EK 1.3.1C

KS1.4.4 Understand that digital effects and animations can be created by using existing software or modified software that includes functionality to implement the effects and animations.

KS1.4.5 Understand that computing enables creative exploration of both real and virtual phenomena. EK 1.3.1E

D2 Problem-Solving Mindset

There are professional characteristics and habits of action that help people create value for society though innovation and problem solving.

O2.1 Describe moments within a process where curiosity, persistence, and the positive aspect of failure played an important role in gaining understanding about a problem or unexpected observation.

KS2.1.1 (2b IWR) Describe difficulties and/or opportunities you encountered and how they were resolved or incorporated.

O2.2 Engage stakeholders in a problem and use their perspectives to shape the course of your development.

KS2.2.1 Identifying programmer and user concerns that affect the solution to problems. EK 5.1.2G

KS2.2.2 Consult and communicate with program users in program development to solve problems. EK 5.1.2H

D3 Problem-Solving Process

A computational problem solving process is an iterative, systematic approach by which a team generates and validates a proposed solution.

O3.1 Apply and describe an iterative, process based on user-centered research to solve a problem.

KS3.1.1 Apply and describe an iterative process used during the development of a solution.

KS3.1.2 Use user-centered research and design techniques to create software solutions. 3A-A-5-5

O3.2 Identify and apply decomposition as a critical step in problem solving.

KS3.2.1 Deconstruct a complex project or problem into smaller discrete modules that can be developed independently, then incorporated together at a later time.

KS3.2.2 Deconstruct a complex problem into simpler parts using predefined constructs (e.g., functions and parameters and/or classes). 3A-A-4-8

O3.3 Explain how people participate in a problem-solving process that scales. LO 7.1.2 [P4]

KS3.3.1 Understand that distributed solutions must scale to solve some problems. EK 7.1.2A

KS3.3.2 Understand that science has been impacted by using scale and “citizen science” to solve scientific problems using home computers in scientific research. EK 7.1.2B

KS3.3.3 Understand that human computation harnesses contributions from many humans to solve problems related to digital data and the Web. EK 7.1.2C

KS3.3.4 Understand that human capabilities are enhanced by digitally enabled collaboration. EK 7.1.2D

KS3.3.5 Understand that some online services use the contributions of many people to benefit both individuals and society. EK 7.1.2E

KS3.3.6 Understand that crowdsourcing offers new models for collaboration, such as connecting people with jobs and businesses with funding. EK 7.1.2F

KS3.3.7 Understand that the move from desktop computers to a proliferation of always-on mobile computers is leading to new applications. EK 7.1.2G

O3.4 (1V) Create a computational artifact using computing tools and techniques to solve a problem. [P2] LO 1.2.2

KS3.4.1 Understand that computing tools and techniques can enhance the process of finding a solution to a problem. EK 1.2.2A

KS3.4.2 Understand that a creative development process for creating computational artifacts can be used to solve problems when traditional or prescribed computing techniques are not effective. EK 1.2.2B

O3.5 Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] LO 1.2.5

KS3.5.1 Understand that the context in which an artifact is used determines the correctness, usability, functionality, and suitability of the artifact. EK 1.2.5A

KS3.5.2 Understand that a computational artifact may have weaknesses, mistakes, or errors depending on the type of artifact. EK 1.2.5B

KS3.5.3 Understand that the functionality of a computational artifact may be related to how it is used or perceived. EK 1.2.5C

KS3.5.4 Understand that the suitability (or appropriateness) of a computational artifact may be related to how it is used or perceived. EK 1.2.5D

D4 Computational Tools and Techniques

Computing involves the application of collaboration tools, programming tools, mathematical principles, and techniques to manage developments.

O4.1 Select and apply appropriate computational tools and techniques to solve a problem or create value for others.

KS4.1.1 Select tools for collaborating for data collection, writing, or programming.

KS4.1.2 Gain understanding of software tools and services while creating computational artifacts. EK 1.2.1B

KS4.1.3 Apply computing tools and techniques to create computational artifacts including, but not limited to, programming integrated development environments (IDEs). EK 1.2.1C

KS4.1.4 Navigate and use unfamiliar documentation and public information to extend the student's own knowledge of a programming language or to achieve a computational approach to solve a problem.

O4.2 Apply a system of version control effectively.

KS4.2.1 Maintain successive versions of a digital product during development.

C2 Technical Knowledge and Skills

Every career field requires technical literacy and career-specific knowledge and skills to support professional practice.

D5 Data

Data and information facilitate the creation of knowledge. Managing and interpreting an overwhelming amount of raw data is part of the foundation of our information society and economy.

O5.1 Find patterns and test hypotheses about digitally processed information to gain insight and knowledge. [P4]

KS5.1.1 Understand that computers are used in an iterative and interactive way when processing digital information to gain insight and knowledge. EK 3.1.1A

KS5.1.2 Understand that digital information can be filtered and cleaned by using computers to process information. EK 3.1.1B

KS5.1.3 Understand that combining data sources, clustering data, and data classification are part of the process if using computers to process information. EK 3.1.1C

KS5.1.5 Understand that patterns can emerge when data is transformed using computational tools. EK 3.1.1E

O5.2 Extract information from data to discover and explain connections or trends. [P1]

KS5.2.1 Understand that large data sets provide opportunities and challenges for extracting information and knowledge. EK 3.2.1A

KS5.2.2 Understand that large data sets provide opportunities for identifying trends, making connections in data, and solving problems. EK 3.2.1B

KS5.2.3 Understand that computing tools facilitate the discovery of connections in information within large data sets. EK 3.2.1C

KS5.2.4 Understand that search tools are essential for efficiently finding information. EK 3.2.1D

KS5.2.6 Understand that software tools, including spreadsheets and databases, help to efficiently organize and find trends in information. EK 3.2.1F (Students are not expected to know specific formulas or options available in spreadsheet or database soft

KS5.2.7 Understand that metadata is data about data. EK 3.2.1G

KS5.2.8 Understand that metadata can be descriptive data about an image, a Web page, or other complex objects. EK 3.2.1H

KS5.2.9 Understand that metadata can increase the effective use of data or data sets by providing additional information about various aspects of that data. EK 3.2.1I

O5.3 Explain the insight and knowledge gained from digitally processed data by using appropriate visualizations, notations, and precise language. [P5]

KS5.3.1 Understand that visualization tools and software can communicate information about data. EK 3.1.3A

KS5.3.2 Understand that tables, diagrams, and textual displays can be used in communicating insight and knowledge gained from data. EK 3.1.3B

KS5.3.3 Understand that summaries of data analyzed computationally can be effective in communicating insight and knowledge gained from digitally represented information. EK 3.1.3C

KS5.3.5 Understand that interactivity with data is an aspect of communicating. EK 3.1.3E

O5.4 Determine how large data sets impact the use of computational processes to discover information and knowledge. [P3]

KS5.4.1 Understand that large data sets include data such as transactions, measurements, texts, sounds, images, and videos. EK 3.2.2A

KS5.4.2 Understand that the storing, processing, and curating of large data sets is challenging. EK 3.2.2B

KS5.4.3 Understand that structuring large data sets for analysis can be challenging. EK 3.2.2C

KS5.4.4 Understand that maintaining privacy of large data sets containing personal information can be challenging. EK 3.2.2D

KS5.4.5 Understand that scalability of systems is an important consideration when data sets are large. EK 3.2.2E

KS5.4.6 Understand that the size or scale of a system that stores data affects how that data set is used. EK 3.2.2F

KS5.4.7 Understand that the effective use of large data sets requires computational solutions. EK 3.2.2G

O5.5 Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information. [P4]

KS5.5.1 Understand that digital data representations involve trade-offs related to storage, security, and privacy concerns. EK 3.3.1A

KS5.5.2 Understand that security concerns engender trade-offs in storing and transmitting information. EK 3.3.1B

KS5.5.3 Understand that there are trade-offs in using lossy and lossless compression techniques for storing and transmitting data. EK 3.3.1C

KS5.5.4 Understand that lossless data compression reduces the number of bits stored or transmitted but allows complete reconstruction of the original data. EK 3.3.1D

KS5.5.5 Understand that lossy data compression can significantly reduce the number of bits stored or transmitted at the cost of being able to reconstruct only an approximation of the original data. EK 3.3.1E

KS5.5.6 Understand that security and privacy concerns arise with data containing personal information. EK 3.3.1F

KS5.5.7 Understand that data is stored in many formats depending on its characteristics (e.g., size and intended use). EK 3.3.1G

KS5.5.8 Understand that the choice of storage media affects both the methods and costs of manipulating the data it contains. EK 3.3.1H

KS5.5.9 Understand that reading data and updating data have different storage requirements. EK 3.3.1I

O5.6 Determine how large data sets impact the use of computational processes to discover information and knowledge. [P3]

KS5.6.1 Understand that large data sets include data such as transactions, measurements, texts, sounds, images, and videos. EK 3.2.2A

KS5.6.2 Understand that the storing, processing, and curating of large data sets is challenging. EK 3.2.2B

KS5.6.3 Understand that structuring large data sets for analysis can be challenging. EK 3.2.2C

KS5.6.4 Understand that maintaining privacy of large data sets containing personal information can be challenging. EK 3.2.2D

KS5.6.5 Understand that scalability of systems is an important consideration when data sets are large. EK 3.2.2E

KS5.6.6 Understand that the size or scale of a system that stores data affects how that data set is used. EK 3.2.2F

KS5.6.7 Understand that the effective use of large data sets requires computational solutions. EK 3.2.2G

KS5.6.8 Understand that analytical techniques to store, manage, transmit, and process data sets change as the size of data sets scale. EK 3.2.2H

D6 Algorithms

Algorithms are used to develop and express solutions to computational problems. Algorithms are fundamental to even the most basic everyday task.

O6.1 Develop an algorithm for implementation in a program. LO 4.1.1 [P2]

KS6.1.1 Understand that sequencing, selection, and iteration are building blocks of algorithms. EK 4.1.1A

KS6.1.2 Understand that sequencing is the application of each step of an algorithm in the order in which the statements are given. EK 4.1.1B

KS6.1.3 Use a Boolean condition or selection to determine which of two parts of an algorithm are used. EK 4.1.1C

KS6.1.4 Use Iteration or repetition of a part of an algorithm until a condition is met or until a specified number of times have been completed. EK 4.1.1D

KS6.1.5 (2c IWR) Combine algorithms to make new algorithms and explain how they function both independently and together. EK 4.1.1E

KS6.1.6 Use existing correct algorithms as building blocks for constructing a new algorithm to help ensure the new algorithm is correct. EK 4.1.1F

KS6.1.7 Identify different algorithms that can be developed to solve the same problem. EK 4.1.1H

KS6.1.8 Develop a new algorithm to solve a problem can yield insight into the problem. EK 4.1.1I

KS6.1.9 Implement and analyze common patterns employing variables and iteration, including “for” loops iterating across a list, value accumulation, and list aggregation.

O6.2 Express an algorithm in a language. LO 4.1.2 [P5]

KS6.2.1 Understand that languages for algorithms include natural language, pseudocode, and visual and textual programming languages. EK 4.1.2A

KS6.2.2 Understand that natural language and pseudocode describe algorithms so that humans can understand them. EK 4.1.2B

KS6.2.3 Understand that algorithms described in programming languages can be executed on a computer. EK 4.1.2C

KS6.2.4 Understand that different languages are better suited for expressing different algorithms. EK 4.1.2D

KS6.2.5 Understand that some programming languages are designed for specific domains and are better for expressing algorithms in those domains. EK 4.1.2E

KS6.2.6 Understand that the language used to express an algorithm can affect characteristics such as clarity or readability but not whether an algorithmic solution exists. EK 4.1.2F

KS6.2.7 Understand that every algorithm can be constructed using only sequencing, selection, and iteration. EK 4.1.2G

KS6.2.8 Understand that nearly all programming languages are equivalent in terms of being able to express any algorithm. EK 4.1.2H

KS6.2.9 Understand that clarity and readability are important considerations when expressing an algorithm in a language. EK 4.1.2I

O6.3 Explain the difference between algorithms that run in a reasonable time and those that do not run in a reasonable time. [P1] (Any discussion of nondeterministic polynomial (NP) is beyond the scope of this course and the AP Exam.)

KS6.3.1 Understand that many problems can be solved in a reasonable time. EK 4.2.1A

KS6.3.2 Understand that reasonable time means that the number of steps the algorithm takes is less than or equal to a polynomial function (constant, linear, square, cube, etc.) of the size of the input. EK 4.2.1B Using nonpolynomial functions to describe relationships between the number of steps required by an algorithm and the input size is beyond the scope of this curse and the AP Exam.

KS6.3.3 Understand that some problems cannot be solved in a reasonable time, even for small input sizes. EK 4.2.1C

O6.6 Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity. [P4]

KS6.6.1 Understand that determining an algorithm’s efficiency is done by reasoning formally or mathematically about the algorithm. EK 4.2.4A

KS6.6.2 Understand that empirical analysis of an algorithm is done by implementing the algorithm and running it on different inputs. EK 4.2.4B

KS6.6.4 Understand that different correct algorithms for the same problem can have different efficiencies. EK 4.2.4D

KS6.6.5 Understand that sometimes, more efficient algorithms are more complex. EK 4.2.4E

KS6.6.6 Understand that finding an efficient algorithm for a problem can help solve larger instances of the problem. EK 4.2.4F

KS6.6.7 Understand that efficiency includes both execution time and memory usage. EK 4.2.4G (Formal analysis of algorithms (Big-O) and formal reasoning using mathematical formulas are beyond the scope of this course and the AP Exam.)

KS6.6.8 Understand that linear search can be used when searching for an item in any list; binary search can be used only when the list is sorted. EK 4.2.4H

D7 Abstraction

Abstraction reduces information and detail to facilitate focus on relevant concepts. It is a process, a strategy, and the result of reducing detail to focus on concepts relevant to understanding and solving problems.

O7.1 Describe the variety of abstractions used to represent data. LO 2.1.1 [P3]

- KS7.1.1 Understand that digital data is represented by abstractions at different levels. EK 2.1.1A
- KS7.1.2 Understand that at the lowest level, all digital data are represented by bits. EK 2.1.1B
- KS7.1.3 Understand that at a higher level, bits are grouped to represent abstractions, including but not limited to numbers, characters, and color. EK 2.1.1C
- KS7.1.4 Understand that number bases, including binary, decimal, and hexadecimal, are used to represent and investigate digital data. EK 2.1.1D
- KS7.1.5 Understand that at one of the lowest levels of abstraction, digital data is represented in binary (base 2) using only combinations of the digits zero and one. EK 2.1.1E (Two's complement conversions are beyond the scope of this course and the AP
- KS7.1.6 Understand that hexadecimal (base 16) is used to represent digital data because hexadecimal representation uses fewer digits than binary. EK 2.1.1F
- KS7.1.7 Understand that numbers can be converted from any base to any other base. EK 2.1.1G

O7.2 Explain how binary sequences are used to represent digital data. [P5]

- KS7.2.1 Understand that a finite representation is used to model the infinite mathematical concept of a number. EK 2.1.2A (Binary representations of scientific notation are beyond the scope of this course and the AP Exam.)
- KS7.2.2 Understand that in many programming languages, the fixed number of bits used to represent characters or integers limits the range of integer values and mathematical operations; this limitation can result in overflow or other errors. EK 2.1.2B (Ra
- KS7.2.3 Understand that in many programming languages, the fixed number of bits used to represent real numbers (as floatingpoint numbers) limits the range of floating-point values and mathematical operations; this limitation can result in round-off and other errors.
- KS7.2.4 Understand that the interpretation of a binary sequence depends on how it is used. EK 2.1.2D
- KS7.2.5 Understand that a sequence of bits may represent instructions or data. EK 2.1.2E
- KS7.2.6 Understand that a sequence of bits may represent different types of data in different contexts. EK 2.1.2F

O7.3 Describe an abstraction used when writing a program or creating other computational artifacts. LO 2.2.1 [P2]

- KS7.3.1 Understand that create an abstraction that generalizes functionality with input parameters that allow reuse. EK 2.2.1C
- KS7.3.2 Understand that represent multiple levels of abstractions, such as constants, expressions, statements, procedures, and libraries. EK 2.2.2A
- KS7.3.3 (2d IWR) Describe how an abstraction is used to manage complexity in a specific program.

O7.4 Develop an abstraction when writing a program or creating other computational artifacts. [P2]

- KS7.4.1 Understand that the process of developing an abstraction involves removing detail and generalizing functionality. EK 2.2.1A
- KS7.4.2 Understand that an abstraction extracts common features from specific examples in order to generalize concepts. EK 2.2.1B
- KS7.4.3 Understand that an abstraction generalizes functionality with input parameters that allow software reuse. EK 2.2.1C (An understanding of the difference between value and reference parameters is beyond the scope of this course and the AP Exam.)

O7.5 Use multiple levels of abstraction to write programs. [P3]

KS7.5.1 Understand that software is developed using multiple levels of abstractions, such as constants, expressions, statements, procedures, and libraries. EK 2.2.2A

KS7.5.2 Understand that being aware of and using multiple levels of abstractions in developing programs help to more effectively apply available resources and tools to solve problems. EK 2.2.2B

O7.6 Identify multiple levels of abstractions that are used when writing programs. [P3]

KS7.6.1 Understand that different programming languages offer different levels of abstraction. EK 2.2.3A (Knowledge of the abstraction capabilities of all programming languages is beyond the scope of this course and the AP Exam.)

KS7.6.2 Understand that high-level programming languages provide more abstractions for the programmer and make it easier for people to read and write a program. EK 2.2.3B

KS7.6.3 Understand that code in a programming language is often translated into code in another (lower-level) language to be executed on a computer. EK 2.2.3C

KS7.6.4 Understand that in an abstraction hierarchy, higher levels of abstraction (the most general concepts) would be placed toward the top and lower-level abstractions (the more specific concepts) toward the bottom. EK 2.2.3D

KS7.6.5 Understand that binary data is processed by physical layers of computing hardware, including gates, chips, and components. EK 2.2.3E

KS7.6.6 Understand that a logic gate is a hardware abstraction that is modeled by a Boolean function. EK 2.2.3F (Memorization of specific gate visual representations is beyond the scope of this course and the AP Exam.)

KS7.6.7 Understand that a chip is an abstraction composed of low-level components and circuits that perform a specific function. EK 2.2.3G

KS7.6.8 Understand that a hardware component can be low level like a transistor or high level like a video card. EK 2.2.3H

KS7.6.9 Understand that hardware is built using multiple levels of abstractions, such as transistors, logic gates, chips, memory, motherboards, special purpose cards, and storage devices. EK 2.2.3I

KS7.6.11 Understand that lower-level abstractions can be combined to make higher-level abstractions, such as short message services (SMS) or email messages, images, audio files, and videos. EK 2.2.3K

D8 Programming

Programming enables problem solving, human expression, and creation of knowledge. Any particular programming language is selected based on appropriateness for a specific project or problem.

O8.1 Creative Expression in Programming - O8.1 (2a IWR) Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. LO 5.1.1 [P2] Or Problem Solving in Programming - O8.1 (2b IWR) Develop a program to solve problems. L

KS8.1.1 Develop programs used in a variety of ways by a wide range of people. EK 5.1.1A

KS8.1.2 Understand that programs developed for creative expression, to satisfy personal curiosity, or to create new knowledge may have visual, audible, or tactile inputs and outputs. EK 5.1.1B

KS8.1.3 Understand that programs developed for creative expression, to satisfy personal curiosity, or to create new knowledge may be developed with different standards or methods than programs developed for widespread distribution. EK 5.1.1C

O8.2 Iteration in Programming - O8.2 Create programs by writing and testing code in a modular, incremental approach.

KS8.2.1 (2b IWR) Describe how an iterative process of program development helps in developing a correct program to solve problems. EK 5.1.2A

KS8.2.3 Incrementally add tested program segments to correct working programs to help create larger correct programs. EK 5.1.2C

KS8.2.4 Adapt or improve existing code.

KS8.2.5 Understand that program documentation helps programmers develop and maintain correct programs to efficiently solve problems. EK 5.1.2D

KS8.2.6 Understand that dcumentation about program components, such as code segments and procedures, helps in developing and maintaining programs. EK 5.1.2E

KS8.2.7 Understand that documentation helps in developing and maintaining programs when working individually or in collaborative programming environments. EK 5.1.2F

KS8.2.8 Understand that program development includes identifying programmer and user concerns that affect the solution to problems. EK 5.1.2G

KS8.2.9 Understand that consultation and communication with program users is an important aspect of program development to solve problems. EK 5.1.2H

KS8.2.10 Understand that a programmer's knowledge and skill affects how a program is developed and how it is used to solve a problem. EK 5.1.2I

KS8.2.11 Understand that a programmer designs, implements, tests, debugs, and maintains programs when solving problems. EK 5.1.2J

O8.3 Algorithms in Programs - O8.3 Explain how programs implement algorithms. LO 5.2.1 [P3]

KS8.3.1 Understand that algorithms are implemented using program instructions that are processed during program execution. EK 5.2.1A

KS8.3.2 Understand that program instructions are executed sequentially. EK 5.2.1B

KS8.3.3 Understand that program instructions may involve variables that are initialized and updated, read, and written. EK 5.2.1C

KS8.3.4 Describe that an understanding of instruction processing and program execution is useful for programming. EK 5.2.1D

KS8.3.5 Understand that program execution automates processes. EK 5.2.1E

KS8.3.6 Understand that processes use memory, a central processing unit (CPU), and input and output. EK 5.2.1F

KS8.3.8 Understand that a process may execute on one or several CPUs. EK 5.2.1H

KS8.3.10 Understand that simple algorithms can solve a large set of problems when automated. EK 5.2.1J

KS8.3.11 Understand that improvements in algorithms, hardware, and software increase the kinds of problems and the size of problems solvable by programming. EK 5.2.1K

O8.4 Abstraction in Programs - O8.4 (2d IWR) Use an abstraction to manage complexity in programs. LO 5.3.1 [P3]

KS8.4.1 Understand that procedures are reusable programming abstractions. EK 5.3.1A

KS8.4.2 Understand that a procedure is a named grouping of programming instructions. EK 5.3.1B

KS8.4.3 Use procedures to reduce the complexity of writing and maintaining programs. EK 5.3.1C

KS8.4.4 Understand that procedures have names and may have parameters and return values. EK 5.3.1D

KS8.4.5 Describe how parameterization can generalize a specific solution. EK 5.3.1E

KS8.4.6 Use parameters to generalize a solution by allowing a procedure to be used instead of duplicated code. EK 5.3.1F

KS8.4.7 Understand that parameters provide different values as input to procedures when they are called in a program. EK 5.3.1G

KS8.4.8 Understand that data abstraction provides a means of separating behavior from implementation. EK 5.3.1H

KS8.4.9 Understand that strings and string operations, including concatenation and some form of substring, are common in many programs. EK 5.3.1I

KS8.4.10 Understand that integers and floating-point numbers are used in programs without requiring understanding of how they are implemented. EK 5.3.1J

KS8.4.11 Understand that lists and list operations, such as add, remove, and search, are common in many programs. EK 5.3.1K

KS8.4.12 Understand that using lists and procedures as abstractions in programming can result in programs that are easier to develop and maintain. EK 5.3.1L

KS8.4.13 Understand that application program interfaces (APIs) and libraries simplify complex programming tasks. EK 5.3.1M

KS8.4.14 Understand that documentation for an API/library is an important aspect of programming. EK 5.3.1N

KS8.4.15 Understand that APIs connect software components, allowing them to communicate. EK 5.3.1O

O8.5 Evaluate the correctness of a program. [P4]

KS8.5.1 Program style can affect the determination of program correctness. EK 5.4.1A

KS8.5.2 Understand that duplicated code can make it harder to reason about a program. EK 5.4.1B

KS8.5.3 Understand that meaningful names for variables and procedures help people better understand programs. EK 5.4.1C

KS8.5.4 Understand that longer code segments are harder to reason about than shorter code segments in a program. EK 5.4.1D

KS8.5.5 Understand that locating and correcting errors in a program is called debugging the program. EK 5.4.1E

KS8.5.6 Understand that knowledge of what a program is supposed to do is required in order to find most program errors. EK 5.4.1F

KS8.5.7 Understand that examples of intended behavior on specific inputs help people understand what a program is supposed to do. EK 5.4.1G

KS8.5.8 Understand that visual displays (or different modalities) of program state can help in finding errors. EK 5.4.1H

KS8.5.9 Understand that programmers justify and explain a program’s correctness. EK 5.4.1I

KS8.5.10 Understand that justification can include a written explanation about how a program meets its specifications. EK 5.4.1J

KS8.5.11 Understand that correctness of a program depends on correctness of program components, including code segments and procedures. EK 5.4.1K

KS8.5.12 Understand that an explanation of a program helps people understand the functionality and purpose of it. EK 5.4.1L

KS8.5.13 Understand that the functionality of a program is often described by how a user interacts with it. EK 5.4.1M

KS8.5.14 Understand that the functionality of a program is best described at a high level by what the program does, not at the lower level of how the program statements work to accomplish this. EK 5.4.1N

O8.6 Mathematical and Logic Concepts in Programming - O8.6 (2c IWR) Employ and describe appropriate mathematical and logical concepts in programming. LO 5.5.1 [P1]

KS8.6.1 Understand that strings and string operations, including concatenation and some form of substring, are common in many programs. EK 5.3.1

KS8.6.2 Recognize that numbers and numerical concepts are fundamental to programming. EK 5.5.1A

KS8.6.3 Understand that integers may be constrained in the maximum and minimum values that can be represented in a program because of storage limitations. EK 5.5.1B

KS8.6.4 Understand that real numbers are approximated by floating-point representations that do not necessarily have infinite precision. EK 5.5.1C

KS8.6.5 Understand that mathematical expressions using arithmetic operators are part of most programming languages. EK 5.5.1D

KS8.6.6 Understand that logical concepts and Boolean algebra are fundamental to programming. EK 5.5.1E

KS8.6.7 Use compound expressions using and, or, and not. EK 5.5.1F

KS8.6.8 Understand that intuitive and formal reasoning about program components using Boolean concepts helps in developing correct programs. EK 5.5.1G

KS8.6.9 Use computational methods such as lists and collections to solve problems. EK 5.5.1H

KS8.6.10 Understand that lists and other collections can be treated as abstract data types (ADTs) in developing programs. EK 5.5.1I

D9 Modeling and Simulation

People use computer programs to process information to gain insight and knowledge.

O9.1 Use models and simulations to represent phenomena. LO 2.3.1 [P3]

KS9.1.1 Understand that models and simulations are simplified representations of more complex objects or phenomena. EK 2.3.1A

KS9.1.2 Understand that models may use different abstractions or levels of abstraction depending on the objects or phenomena being posed. EK 2.3.1B

KS9.1.3 Understand that models often omit unnecessary features of the objects or phenomena that are being modeled. EK 2.3.1C

KS9.1.4 Understand that simulations mimic real-world events without the cost or danger of building and testing the phenomena in the real world. EK 2.3.1D

O9.2 Use models and simulations to formulate, refine, and test hypotheses. [P3]

KS9.2.1 Understand that models and simulations facilitate the formulation and refinement of hypotheses related to the objects or phenomena under consideration. EK 2.3.2A

KS9.2.2 Understand that hypotheses are formulated to explain the objects or phenomena being modeled. EK 2.3.2B

KS9.2.3 Understand that hypotheses are refined by examining the insights that models and simulations provide into the objects or phenomena. EK 2.3.2C

KS9.2.4 Understand that the results of simulations may generate new knowledge and new hypotheses related to the phenomena being modeled. EK 2.3.2D

KS9.2.5 Understand that simulations allow hypotheses to be tested without the constraints of the real world. EK 2.3.2E

KS9.2.6 Understand that simulations can facilitate extensive and rapid testing of models. EK 2.3.2F

KS9.2.7 Understand that the time required for simulations is impacted by the level of detail and quality of the models and the software and hardware used for the simulation. EK 2.3.2G

KS9.2.8 Understand that rapid and extensive testing allows models to be changed to accurately reflect the objects or phenomena being modeled. EK 2.3.2H

D10 The Internet

The internet pervades modern computing. The internet and the systems built on it have had a profound impact on society. Computer networks support communication and collaboration.

O10.1 Explain characteristics of the internet and the systems built on it. LO 6.2.1 [P5]

KS10.1.1 Understand that the Internet connects devices and networks all over the world. EK 6.1.1A

KS10.1.2 Understand that an end-to-end architecture facilitates connecting new devices and networks on the Internet. EK 6.1.1B

KS10.1.3 Understand that sevices and networks that make up the Internet are connected and communicate using addresses and protocols. EK 6.1.1C

KS10.1.4 Understand that the Internet and the systems built on it facilitate collaboration. EK 6.1.1D

KS10.1.5 Understand that connecting new devices to the Internet is enabled by assignment of an Internet protocol (IP) address. EK 6.1.1E

KS10.1.6 Understand that the Internet is built on evolving standards, including those for addresses and names. EK 6.1.1F (Specific details of any particular standard for addresses are beyond the scope of this course and the AP Exam.)

KS10.1.7 Understand that the domain name system (DNS) translates domain names to IP addresses. EK 6.1.1G

KS10.1.8 Understand that the number of devices that could use an IP address has grown so fast that a new protocol (IPv6) has been established to handle routing of many more devices. EK 6.1.1H

KS10.1.9 Understand that standards such as hypertext transfer protocol (HTTP), IP, and simple mail transfer protocol (SMTP) are developed and overseen by the Internet Engineering Task Force (IETF). EK 6.1.1I

O10.2 Explain characteristics of the Internet and the systems built on it. [P5]

KS10.2.1 Understand that the Internet and the systems built on it are hierarchical and redundant. EK 6.2.1A

KS10.2.2 Understand that the domain name syntax is hierarchical. EK 6.2.1B

KS10.2.3 Understand that IP addresses are hierarchical. EK 6.2.1C

KS10.2.4 Understand that routing on the Internet is fault tolerant and redundant. EK 6.2.1D

O10.3 Explain how the characteristics of the Internet influence the systems built on it. [P4]

KS10.3.1 Understand that hierarchy and redundancy help systems scale. EK 6.2.2A

KS10.3.2 Understand that the redundancy of routing (i.e., more than one way to route data) between two points on the Internet increases the reliability of the Internet and helps it scale to more devices and more people. EK 6.2.2B

KS10.3.3 Understand that hierarchy in the DNS helps that system scale. EK 6.2.2C

KS10.3.4 Understand that interfaces and protocols enable widespread use of the Internet. EK 6.2.2D

KS10.3.5 Understand that open standards fuel the growth of the Internet. EK 6.2.2E

KS10.3.6 Understand that the Internet is a packet-switched system through which digital data is sent by breaking the data into blocks of bits called KS10.3.7

KS10.3.7 Understand that standards for packets and routing include transmission control protocol/Internet protocol (TCP/IP). EK 6.2.2G

KS10.3.8 Understand that standards for sharing information and communicating between browsers and servers on the Web include HTTP and secure sockets layer/transport layer security (SSL/TLS). EK 6.2.2H (Understanding the technical aspects of how SSL/TLS w

KS10.3.10 Understand that the bandwidth of a system is a measure of bit rate — the amount of data (measured in bits) that can be sent in a fixed amount of time. EK 6.2.2J

KS10.3.11 Understand that the latency of a system is the time elapsed between the transmission and the receipt of a request. EK 6.2.2K

O10.4 Explain how computing innovations affect communication, interaction, and cognition

KS10.4.1 Understand that email, SMS, and chat have fostered new ways to communicate and collaborate. EK 7.1.1A

KS10.4.2 Understand that video conferencing and video chat have fostered new ways to communicate and collaborate. EK 7.1.1B

KS10.4.4 Understand that cloud computing fosters new ways to communicate and collaborate. EK 7.1.1D

KS10.4.5 Understand that widespread access to information facilitates the identification of problems, development of solutions, and dissemination of results. EK 7.1.1E

KS10.4.6 Understand that public data provides widespread access and enables solutions to identified problems. EK 7.1.1F

KS10.4.7 Understand that search trends are predictors. EK 7.1.1G

KS10.4.8 Understand that social media, such as blogs and Twitter, have enhanced dissemination. EK 7.1.1H

KS10.4.9 Understand that Global Positioning System (GPS) and related technologies have changed how humans travel, navigate, and find information related to geolocation. EK 7.1.1I

KS10.4.10 Understand that sensor networks facilitate new ways of interacting with the environment and with physical systems. EK 7.1.1J

KS10.4.12 Understand that computing contributes to many assistive technologies that enhance human capabilities. EK 7.1.1L

KS10.4.13 Understand that the Internet and the Web have enhanced methods of and opportunities for communication and collaboration. EK 7.1.1M

KS10.4.14 Understand that the Internet and the Web have changed many areas, including e-commerce, health care, access to information and entertainment, and online learning. EK 7.1.1N

KS10.4.15 Understand that the Internet and the Web have impacted productivity, positively and negatively, in many areas. EK 7.1.1O

D11 Cybersecurity

Cybersecurity is an important concern for the internet and the systems built on it.

O11.1 Identify existing cybersecurity concerns and potential options to address these issues with the internet and the systems built on it. LO 6.3.1 [P1]

KS11.1.2 Understand that the DNS was not designed to be completely secure. EK 6.3.1B

KS11.1.3 Understand that implementing cybersecurity has software, hardware, and human components. EK 6.3.1C

KS11.1.4 Understand that cyberwarfare and cybercrime have widespread and potentially devastating effects. EK 6.3.1D

KS11.1.5 Understand that distributed denial-of-service attacks (DDoS) compromise a target by flooding it with requests from multiple systems. EK 6.3.1E

KS11.1.6 Understand that phishing, viruses, and other attacks have human and software components. EK 6.3.1F

KS11.1.7 Understand that antivirus software and firewalls can help prevent unauthorized access to private data. EK 6.3.1G

KS11.1.8 Understand that cryptography is essential to many models of cybersecurity. EK 6.3.1H

KS11.1.9 Understand that cryptography has a mathematical foundation. EK 6.3.1I (Specific mathematical functions used in cryptography are beyond the scope of this course and the AP Exam.)

KS11.1.10 Understand that open standards help ensure cryptography is secure. EK 6.3.1J

KS11.1.11 Understand that symmetric encryption is a method of encryption involving one key for encryption and decryption. EK 6.3.1K

KS11.1.12 Understand that public key encryption, which is not symmetric, is an encryption method that is widely used because of the functionality it provides. EK 6.3.1L (The mathematical methods used in public key cryptography are beyond the scope of thi

KS11.1.13 Understand that certificate authorities (CAs) issue digital certificates that validate the ownership of encrypted keys used in secured communications and are based on a trust model. EK 6.3.1M

O11.2 Identify user actions that strengthen the security of a networked computing system.

KS11.2.1 Describe secure practices related to passwords, antivirus software, software updates, and posting content online.

KS11.2.2 Identify the unique circumstances in which penetration testing is legal and ethical.

C3 Professional Practices and Communication

Professional practice is guided by professional ethics and standards and requires effective communication and collaboration.

D12 Social Impacts of Computing

Cybersecurity affects economic, environmental, and societal contexts.

O12.1 Explain the connections between computing and real-world contexts, including economic, social, and cultural contexts. LO 7.4.1

KS12.1.1 Understand that the innovation and impact of social media and online access varies in different countries and in different socioeconomic groups. EK 7.4.1A

KS12.1.2 Describe how mobile, wireless, and networked computing has an impact on innovation throughout the world. EK 7.4.1B

KS12.1.3 Describe how the global distribution of computing resources raises issues of equity, access, and power. EK 7.4.1C

KS12.1.4 Understand that groups and individuals are affected by the “digital divide”—differing access to computing and the internet based on socioeconomic or geographic characteristics. EK 7.4.1D

KS12.1.5 Understand that networks and infrastructure are supported by both commercial and governmental initiatives. EK 7.4.1E

O12.2 Analyze the beneficial and harmful effects of computing. [P4]

KS12.2.1 Understand that innovations enabled by computing raise legal and ethical concerns. EK 7.3.1A

KS12.2.2 Understand that commercial access to music and movie downloads and streaming raises legal and ethical concerns. EK 7.3.1B

KS12.2.3 Understand that access to digital content via peer-to-peer networks raises legal and ethical concerns. EK 7.3.1C

KS12.2.4 Understand that both authenticated and anonymous access to digital information raise legal and ethical concerns. EK 7.3.1D

KS12.2.5 Understand that commercial and governmental censorship of digital information raise legal and ethical concerns. EK 7.3.1E

KS12.2.6 Understand that open source and licensing of software and content raise legal and ethical concerns. EK 7.3.1F

KS12.2.7 Understand that privacy and security concerns arise in the development and use of computational systems and artifacts. EK 7.3.1G

KS12.2.8 Understand that aggregation of information, such as geolocation, cookies, and browsing history, raises privacy and security concerns. EK 7.3.1H

KS12.2.10 Understand that technology enables the collection, use, and exploitation of information about, by, and for individuals, groups, and institutions. EK 7.3.1J

KS12.2.11 Understand that people can have instant access to vast amounts of information online; accessing this information can enable the collection of both individual and aggregate data that can be used and collected. EK 7.3.1K

KS12.2.12 Understand that commercial and governmental curation of information may be exploited if privacy and other protections are ignored. EK 7.3.1L

KS12.2.14 Understand that widespread access to digitized information raises questions about intellectual property. EK 7.3.1N

KS12.2.15 Understand that creation of digital audio, video, and textual content by combining existing content has been impacted by copyright concerns. EK 7.3.1O

KS12.2.16 Understand that the Digital Millennium Copyright Act (DMCA) has been a benefit and a challenge in making copyrighted digital material widely available. EK 7.3.1P

KS12.2.17 Understand that open source and free software have practical, business, and ethical impacts on widespread access to programs, libraries, and code. EK 7.3.1Q

D13 Career Awareness

Today computing impacts almost all careers. There are career specializations within computer science such as software development, security, network, and systems administration.

O13.1 Describe career paths within the computing specialties.

KS13.1.1 Describe a variety of careers within the computing specialties.

KS13.1.2 Recognize the education and credentialing requirements for careers within computing specialties.

KS13.1.3 Demonstrate the initiative and independent learning required to stay current with evolving technology and career needs.

O13.2 Explain how computing has impacted innovation in other fields. LO 7.2.1 [P1]

KS13.2.1 Understand that machine learning and data mining have enabled innovation in medicine, business, and science. EK 7.2.1A

KS13.2.2 Understand that scientific computing has enabled innovation in science and business. EK 7.2.1B

KS13.2.3 Understand that computing enables innovation by providing the ability to access and share information. EK 7.2.1C

KS13.2.4 Understand that open access and Creative Commons have enabled broad access to digital information. EK 7.2.1D

KS13.2.5 Understand that open and curated scientific databases have benefited scientific researchers. EK 7.2.1E

KS13.2.6 Understand that Moore's law has encouraged industries that use computers to effectively plan future research and development based on anticipated increases in computing power. EK 7.2.1F

KS13.2.7 Understand that advances in computing as an enabling technology have generated and increased the creativity in other fields. EK 7.2.1G

D14 Professionalism and Ethics

Computing professionals must make decisions regularly regarding their professional and social conduct when collaborating with developers and engaging with users to get feedback.

O14.1 Abide by professional standards when creating value for people and society.

KS14.1.1 Create and maintain a secure professional identity for accessing IDEs and accessing computer science communities.

KS14.1.2 Provide rationales for all ethical decisions.

KS14.1.3 Engage others with respect and for thought.

O14.2 Access, manage, and attribute information using effective strategies. LO 7.5.1 [P1]

KS14.2.1 Understand that online databases and libraries catalog and house secondary and some primary sources. EK 7.5.1A

KS14.2.2 Use advance search tools, Boolean logic, and key words to refine the search focus and/or limit search results based on a variety of factors (e.g., data, peer-review status, type of publication). EK 7.5.1B

KS14.2.3 Understand that plagiarism is a serious offense that occurs when a person presents another's ideas or words as his or her own. Plagiarism may be avoided by accurately acknowledging sources. EK 7.5.1C

O14.3 Consider accessibility and equity when designing products, creating solutions, and collaborating with others.

KS14.3.1 Describe and give examples of universal design strategies that increase accessibility to computational resources.

KS14.3.2 Explain how diversity on development teams is essential for producing outcomes that serve a diverse audience.

O14.4 Evaluate online and print sources for appropriateness and credibility. LO 7.5.2 [P5]

KS14.4.1 Evaluate the credibility of a source by considering reputation and credentials of the author(s), publisher(s), site owner(s), and/or sponsor(s). EK 7.5.2A

KS14.4.2 Evaluate the relevancy of information from a source and if it supports an appropriate claim or the purpose of the investigation. EK 7.5.2B

D15 Collaboration

Diverse perspectives, good interpersonal relationships, and effective collaboration strategies generate the most robust and innovative solutions.

O15.1 Collaborate when processing information to gain insight and knowledge. LO 3.1.2 [P6

KS15.1.1 Understand that collaboration is an important part of solving data-driven problems. EK 3.1.2A

KS15.1.2 Understand that collaboration facilitates solving computational problems by applying multiple perspectives, experiences, and skill sets. EK 3.1.2B

KS15.1.3 Understand that communication between participants working on data-driven problems gives rise to enhanced insights and knowledge. EK 3.1.2C

KS15.1.4 Understand that collaborating face-to-face and using online collaborative tools can facilitate processing information to gain insight and knowledge. EK 3.1.2E

KS15.1.5 Understand that collaborating face-to-face and using online collaborative tools can facilitate processing information to gain insight and knowledge. EK 3.1.2E

KS15.1.6 Understand that investigating large data sets collaboratively can lead to insight and knowledge not obtained when working alone. EK 3.1.2F

O15.2 Collaborate to develop a program. LO 5.1.3 [P6]

KS15.2.1 Understand that a collaboratively created computational artifact reflects effort by more than one person. EK 1.2.4A

KS15.2.2 Understand that effective collaborative teams consider the use of online collaborative tools. EK 1.2.4B

KS15.2.3 Understand that effective collaborative teams practice interpersonal communication, consensus building, conflict resolution, and negotiation. EK 1.2.4C

KS15.2.4 Understand that effective collaboration strategies enhance performance. EK 1.2.4D

KS15.2.5 Understand that collaboration facilitates the application of multiple perspectives (including sociocultural perspectives) and diverse talents and skills in developing computational artifacts. EK 1.2.4E

KS15.2.6 Understand that a collaboratively created computational artifact can reflect personal expressions of ideas. EK 1.2.4F

O15.3 Apply project management strategies effectively as part of a team.

KS15.3.1 Understand that collaboration can decrease the size and complexity of tasks required of individual programmers. EK 5.1.3A

KS15.3.2 Understand that collaboration facilitates multiple perspectives in developing ideas for solving problems by programming. EK 5.1.3B

KS15.3.3 Understand that collaboration in the iterative development of a program requires different skills than developing a program alone. EK 5.1.3C

KS15.3.4 Understand that collaboration can make it easier to find and correct errors when developing programs. EK 5.1.3D

D16 Communication

Computing professionals must be able to explain and justify the design and appropriateness of their computational choices, and analyze and describe both computational artifacts and the results or behaviors of such artifacts.

O16.1 Communicate ideas, processes, and products to optimize audience perception and understanding.

KS16.1.1	Create program documentation that helps programmers develop and maintain correct programs to efficiently solve problems. EK 5.1.2D
KS16.1.2	Create documentation about program components, such as code segments and procedures, that helps in developing and maintaining programs. EK 5.1.2E
KS16.1.3	Create documentation that helps in developing and maintaining programs when working individually or in collaborative programming environments. EK 5.1.2F
KS16.1.4	Summarize the purpose of a computational artifact.
KS16.1.5	(2b IWR) Communicate which portions of a program you developed independently and which were created collaboratively.