**Sources: AP CSP CED -> code.org AP CSP -> BootstrapWorld Data -> -CMU Academy**

* Use correct terminology, vocabulary and appropriate language to communicate effectively in the workplace
* Select and safely use appropriate tools, supplies, and equipment for a specific task or set of tasks.
* Employ effective time and project management strategies to complete work efficiently and proficiently.
* Apply math concepts, including measurement, operations, and higher mathematics to relevant applications and specific tasks.
* Demonstrate awareness strategies to safely work in a variety of workspaces and locations.
* Explore careers within the cluster to include developing individual career documents.
* **Algorithms and Programming**: The techniques and strategies used to make sense of problems and persevere in solving them by developing algorithms to implement the solutions in several programming languages following the software design life cycle.
* **Networks and the Internet**:
* **Computing Systems**: An understanding of the physical components and software that make up a computing system which communicate and process information in digital form, along with practices and methodology for troubleshooting issues in those systems.
* **Data and Analysis**: The concepts, practices and process of data collection, resource management, applying statistical and graphical techniques to different types of data in order to discover useful information that can communicate storytelling and to inform decision-making.
* **Cybersecurity**:
* Impacts of computing? (from K12CS)

From AP CSP CED

Computational Solution Design—Design and evaluate computational solutions for a purpose.

Algorithms and Program Development—Develop and implement algorithms.

Abstraction in Program Development —Develop programs that incorporate abstractions.

Code Analysis —Evaluate and test algorithms and programs.

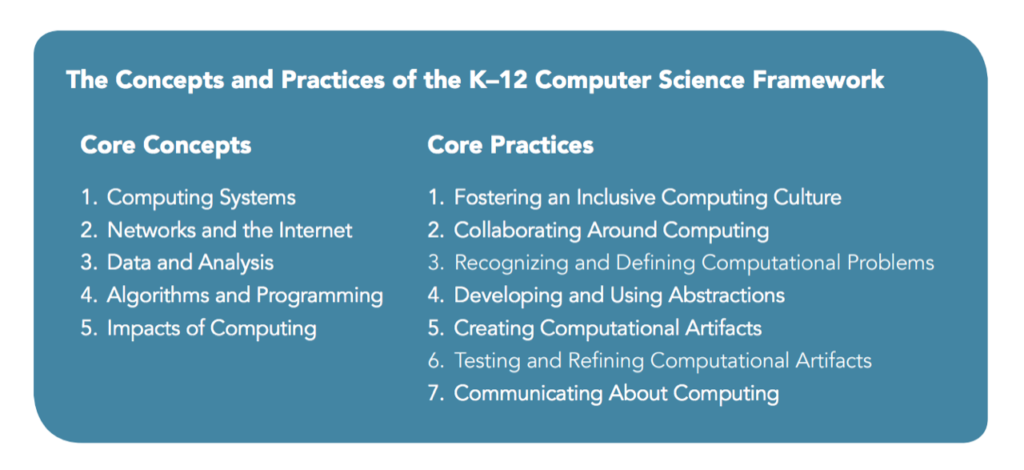
Computing Innovations —Investigate computing innovations.

Responsible Computing—Contribute to an inclusive, safe, collaborative, and ethical computing culture.

Covers these areas of ACM Framework:

* “Networking and Communication”,
* ”Parallel and Distributed Computing, ”
* “Software Development Fundamentals, ”
* “Programming Languages, ”
* “Architecture and Organization, ”
* “Computational Science, ”
* “Information Assurance and Security, ”
* ”Algorithms and Complexity.”

K12CS:



**AP CSP computational thinking practices:**

* Computational Solution Design: Design and evaluate computational solutions for a purpose.
  + I can  Investigate the situation, context, or task.
  + I can  Determine and design an appropriate method or approach to achieve the purpose.
  + I can  Explain how collaboration affects the development of a solution.
  + I can  Evaluate solution options.
* Algorithms and Program Development: Develop and implement algorithms.
  + I can  Represent algorithmic processes without using a programming language.
  + I can  Implement and apply an algorithm.
* Abstraction in Program Development : Develop programs that incorporate abstractions.
  + I can  Generalize data sources through variables.
  + I can  Use abstraction to manage complexity in a program.
  + I can  Explain how abstraction manages complexity.
* Code Analysis : Evaluate and test algorithms and programs
  + I can  Explain how a code segment or program functions.
  + I can  Determine the result of code segments.
  + I can  Identify and correct errors in algorithms and programs, including error discovery through testing.
* . Computing Innovations: Investigate computing innovations.
  + I can  Explain how computing systems work.
  + I can  Explain how knowledge can be generated from data.
  + I can  Describe the impact of a computing innovation.
  + I can  Describe the impact of gathering data.
  + I can  Evaluate the use of computing based on legal and ethical factors.
* Responsible Computing: Contribute to an inclusive, safe, collaborative, and ethical computing culture.
  + I can  Collaborate in the development of solutions.
  + I can Use safe and secure methods when using computing devices.
  + I can  Acknowledge the intellectual property of others.

Big Ideas:

* BIG IDEA 1(10-13%): (Code.org AP CSP) CREATIVE DEVELOPMENT (CRD) When developing computing innovations, developers can use a formal, iterative  **design process**  or a less rigid process of experimentation. While using either approach, developers will encounter phases of investigating and reflecting, designing, prototyping, and testing. Additionally, **collaboration** is an important tool at any phase of development, because considering multiple perspectives allows for improvement of innovations.

ESSENTIAL QUESTIONS CRD-1

* + § How has working collaboratively with other students improved an overall project? § What are some ways you can collect additional feedback on your program to use for improvements?
  + CRD-2 § What are some ways you currently plan your work before starting a project? § What apps or programs have you stopped using because you didn’t like the design of how you interacted with it?
  + I can Explain how computing innovations are improved through collaboration.
  + I can Explain how computing innovations are developed by groups of people.
  + I can Demonstrate effective interpersonal skills during collaboration.
  + I can Describe the purpose of a computing innovation.
  + I can Explain how a program or code segment functions.
  + I can Identify input(s) to a program.
  + I can Identify output(s) produced by a program.
  + I can Develop a program using a development process.
  + I can Design a program and its user interface.
  + I can Describe the purpose of a code segment or program by writing documentation.
  + I can Acknowledge code segments used from other sources.
  + I can For errors in an algorithm or program: a. Identify the error.   b. Correct the error.
  + I can Identify inputs and corresponding expected outputs or behaviors that can be used to check the correctness of an algorithm or program.
* BIG IDEA 2 (17-22%): DATA (DAT)(Bootstrap World Data) Data are central to computing innovations because they communicate initial conditions to programs and represent new knowledge. Computers consume data, transform data, and produce new data, allowing users to create new information or knowledge to solve problems through the interpretation of those data. Computers store data digitally, which means that the data must be manipulated in order to be presented in a useful way to the user.
  + ESSENTIAL QUESTIONS DAT-1
    - § How can we use 1s and 0s t o represent something complex like a video of the marching band playing a song?
    - DAT-2 § How can you predict the attendance at a school e vent using data gathered from social media? § When is it more appropriate to use a computer to analyze data than to complete the analysis by hand?

I can Explain how data can be represented using bits.

I can Explain the consequences of using bits to represent data.

I can For binary numbers: a. Calculate the binary (base 2) equivalent of a positive integer (base 10) and vice versa.  2.B b. Compare and order binary numbers.

I can Compare data compression algorithms to determine which is best in a particular context.

I can Describe what information can be extracted from data.

I can Describe what information can be extracted from metadata.

I can Identify the challenges associated with processing data.

I can Extract information from data using a program.

I can Explain how programs can be used to gain insight and knowledge from data.

* BIG IDEA 3 (30-35%): ALGORITHMS AND PROGRAMMING (AAP) (CMU Academy/Python & code.org APCSP)Programmers integrate algorithms and abstraction to create programs for creative purposes and to solve problems. Using multiple program statements in a specified order, making decisions, and repeating the same process multiple times are the building blocks of programs. Incorporating elements of abstraction—by breaking problems down into interacting pieces, each with their own purpose—makes writing complex programs easier. Programmers need to think algorithmically and use abstraction to define and interpret processes that are used in a program.
* ESSENTIAL QUESTIONS
  + AAP-1 § How can we store data in a program to solve problems?
  + AAP-2 § What might happen if you completed the steps in your regular morning routine to get ready and go to school in a different order? How might the reordering affect the decisions you make each morning?
  + AAP-3 § How do video games group the diff erent actions for a player based on what key is pressed on the keyboard or controller? How do apps group different actions together based on user interaction, such as pressing buttons?
  + AAP-4 § What types of problems can be solved more easily with a computer, and what types can be solved more easily without a computer? Why?
* Topic 3.1: Variables and assignments
  + I can Represent a value with a variable.
  + I can Determine the value of a variable as a result of an assignment.
* Topic 3.2: Data Abstraction
  + I can Represent a list or string using a variable.
  + I can For data abstraction: a. Develop data abstraction using lists to store multiple elements.  b. Explain how the use of data abstraction manages complexity in program code.
* Topic 3.3: Mathematical Expressions
  + I can Express an algorithm that uses sequencing without using a programming language.
  + I can Represent a step-by-step algorithmic process using sequential code statements.
  + I can Evaluate expressions that use arithmetic operators.
* Topic 3.4: Strings
  + I can evaluate expressions that manipulate strings
* Topic 3.5: Boolean Expressions
  + I can, For relationships between Boolean values: a. Write expressions using logical operators.  2.B b. Evaluate expressions that use logic operators.
  + I can For relationships between two variables, expressions, or values: a. Write expressions using relational operators.  b. Evaluate expressions that use relational operators.

Topic 3.6: Conditionals

* + I can Express an algorithm that uses sequencing without using a programming language.
  + I can Represent a step-by-step algorithmic process using sequential code statements.
  + I can Evaluate expressions that use arithmetic operators.
  + I can Evaluate expressions that manipulate strings.
  + I can For relationships between two variables, expressions, or values: a. Write expressions using relational operators.  b. Evaluate expressions that use relational operators.
  + I can For relationships between Boolean values: a. Write expressions using logical operators.  b. Evaluate expressions that use logic operators.
* Topic 3.7: nested conditionals
  + I can For nested selection: a. Write nested conditional statements.  b. Determine the result of nested conditional statements.
* Topic 3.8: Iteration
  + I can Express an algorithm that uses iteration without using a programming language.
  + I can For iteration: a. Write iteration statements. b. Determine the result or side effect of iteration statements.
* Topic 3.9: Developing Algorithms
  + I can Compare multiple algorithms to determine if they yield the same side effect or result.
  + I can For algorithms: a. Create algorithms.   b. Combine and modify existing algorithms.
* Topic 3.10: Lists
  + For list operations: a. Write expressions that use list indexing and list procedures.  b. Evaluate expressions that use list indexing and list procedures.
* BIG IDEA 4 (11-15%): COMPUTING SYSTEMS AND NETWORKS (CSN) Computer systems and networks are used to transfer data. One of the largest and most commonly used networks is the Internet. Through a series of protocols, the Internet can be used to send and receive information and ideas throughout the world. Transferring and processing information can be slow when done on a single computer, but leveraging multiple computers to do the work at the same time can significantly shorten the time it takes to complete tasks or solve problems.
* BIG IDEA 5 (21-26%): IMPACT OF COMPUTING (IOC) Computers and computing have revolutionized our lives. To use computing safely and responsibly, we need to be aware of privacy, security, and ethical issues. As programmers, we need to understand the potential impacts of our programs and be responsible for the consequences. As computer users, we need to understand any potential beneficial or harmful effects and how to protect ourselves and our privacy when using a computer.

Debugging/Testing

ESSENTIAL KNOWLEDGE

CRD-2.I.1 A logic error is a mistake in the algorithm or program that causes it to behave incorrectly or unexpectedly.

CRD-2.I.2 A syntax error is a mistake in the program where the rules of the programming language are not followed.

CRD-2.I.3 A run-time error is a mistake in the program that occurs during the execution of a program. Programming languages define their own run-time errors. CRD-2.I.4 An overflow error is an error that occurs when a computer attempts to handle a number that is outside of the defined range of values.

CRD-2.I.5 The following are effective ways to find and correct errors: § test cases § hand tracing § visualizations § debuggers § adding extra output stat ement(s) Identify inputs and corresponding expected outputs or behaviors that can be used to check the correctness of an algorithm or program.

Code.org alignment to AP:

Use College Board Topic questions as Formative Assessments

| **Unit 1: Digital Information** | **Topic 2.1** Binary Numbers |
| --- | --- |
| **Topic 2.2** Data Compression |
| **Topic 5.5** Legal and Ethical Concerns   * **Note**: A few of the EKs in this topic are not covered until Unit 2 and 10 of the course. Since the majority of the EKs relate to Intellectual Property, we recommend you evaluate student learning of this topic in Unit 1 and return to this topic later in the course. |

| **Unit 2: The Internet** | **Topic 4.1** The Internet |
| --- | --- |
| **Topic 4.2** Fault Tolerance |
| **Topic 5.2** Digital Divide. |

\*\*\* Use CMU Academy for Units 3,4,5 \*\*\*\*\*\*

| **Unit 3: Intro to App Design** | **Topic 1.1** Collaboration   * **Note**: Students will continue to use collaborative practices throughout the course but many explicit skills and ideas will have been introduced in this unit. We recommend you use Topic-based resources in this unit and return to them throughout the course. |
| --- | --- |
| **Topic 1.2** Program Function and Purpose   * **Note**: While the core ideas of this topic are covered in Unit 3, students continue to develop an understanding of these ideas throughout the programming units, the Create PT, and even into Unit 10. We recommend you initially cover these topics here and depending on student performance return to them throughout following programming units. |
| **Topic 1.3** Program Design and Development   * **Note**: This topic is almost entirely covered in Unit 3 but students return to it throughout programming units. |

| **Unit 4: Variables, Conditionals, and Functions** | **Topic 1.4** Identifying and Correcting Errors   * **Note**: Students learn debugging practices in Unit 3 and continue to practice them throughout programming units. We recommend you initially use topic resources here and return to them later if you deem it necessary. |
| --- | --- |
| **Topic 3.1** Variables and Assignment |
| **Topic 3.3** Mathematical Expression |
| **Topic 3.5** Boolean Expressions |
| **Topic 3.6** Conditionals |
| **Topic 3.7** Nested Conditionals |
| **Topic 3.15** Random Values   * **Note**: While students are introduced to random values in Unit 3, we recommend you wait to use resources for this topic until Unit 4 when students have more experience programming expressions with random values |

| **Unit 5: Lists, Loops, and Traversals** | **Topic 3.2** Data Abstraction |
| --- | --- |
| **Topic 3.4** Strings |
| **Topic 3.8** Iteration |
| **Topic 3.10** Lists |
| **Topic 3.16** Simulations |

\*\* Use code.org for Unit 6\*\*\*

| **\*** | **Topic 3.9** Developing Algorithms   * **Note**: Some concepts will have been covered in previous units but we believe this to be the best moment to use these topic resources. |
| --- | --- |
|
|
|
| **Topic 4.3** Parallel and Distributed Computing |

\*\*\*\* Use CMU Academy for Units 7&8

| **Unit 7: Parameters, Return, and Libraries** | **Topic 3.12** Calling Procedures |
| --- | --- |
| **Topic 3.13** Developing Procedures |
| **Topic 3.14** Libraries |

| **Unit 8: Create PT Prep** | **No topics covered** |
| --- | --- |

\*\*\*\*use Bootstrap for Unit 9\*\*\*\*\*

| **Unit 9: Data** | **Topic 2.3** Extracting Information from Data |
| --- | --- |
| **Topic 2.4** Using Programs with Data |
| **Topic 5.3** Computing Bias |
| **Topic 5.4** Crowdsourcing |

\*\*\*Use CodeHS cyber and/or Cyberstart for Cyber. Can also cover unit 10 code.org\*\*\*\*\*

| **Unit 10: Cybersecurity and Global Impacts** | **Topic 5.1** Beneficial and Harmful Effects |
| --- | --- |
| **Topic 5.6** Safe Computing |