The College Board

Computer Science: Principles

Computational Thinking Practices

Big Ideas, Key Concepts, and Supporting Concepts

Computer Science: Principles is a pilot course under development. It is not an official Advanced Placement course currently being offered by the College Board.

This document is based upon work supported by the National Science Foundation, grant CNS-0938336. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Computer Science: Principles

Computational Thinking Practices

1. **Connecting Computing**
    a. Identification of impacts of computing.
    b. Description of connections between people and computing.
    c. Explanation of connections between computing concepts.

2. **Developing computational artifacts**
    a. Creation of an artifact with a practical, personal, or societal intent.
    b. Selection of appropriate techniques to develop a computational artifact.
    c. Use of appropriate algorithmic and information-management principles.

3. **Abstracting**
    a. Explanation of how data, information, or knowledge are represented for computational use.
    b. Explanation of how abstractions are used in computation or modeling.
    c. Identification of abstractions.
    d. Description of modeling in a computational context.

4. **Analyzing problems and artifacts**
    a. Evaluation of a proposed solution to a problem.
    b. Location and correction of errors.
    c. Explanation of how an artifact functions.
    d. Justification of appropriateness and correctness.

5. **Communicating**
    a. Explanation of the meaning of a result in context.
    b. Description using accurate and precise language, notation, or visualizations.
    c. Summary of purpose.

6. **Collaborating**
    a. Collaboration of participants in solving a computational problem.
    b. Collaboration of participants in producing an artifact.
    c. Collaboration at a large scale.

Computer Science: Principles
Big Ideas, Key Concepts, Supporting Concepts

I. **Creativity**: Computing is a creative activity.

A. Computing fosters the creation of artifacts.
   1. Computing enables people to create digitally—including creating knowledge, tools, expressions of ideas, and solutions to problems.
   2. Computing enables people to translate intention into digital artifacts.

B. Computing fosters creative expression.
   1. Computing extends traditional forms of human expression and experience.
   2. Computing fosters the creation of new forms of expression.
   3. Computing enables creative exploration that informs and inspires.

C. Programming is a creative process.
   1. Some programs are developed to satisfy personal curiosity or for creative expression.
   2. Some programs are developed to solve problems, develop new knowledge, or help people, organizations, or society.

II. **Abstraction**: Abstraction reduces information and detail to facilitate focus on relevant concepts.

A. A combination of abstractions built upon binary sequences can be used to represent all digital data.
   1. The interpretation of a binary sequence depends on how it is used (e.g., instruction, number, text, sound, or image).
   2. A finite representation is used to model the infinite mathematical concept of a number.
   3. Number bases, including binary and decimal, are abstractions used for reasoning about digital data.

B. Multiple levels of abstraction are used in computation.
   1. Binary data is processed by physical layers of computing hardware, including gates, chips, and components.
   2. Programming languages, from low to high level, are used in developing software.
   3. Applications and systems are designed, developed, and analyzed using levels of hardware, software, and conceptual abstractions.

C. Models and simulations use abstraction to raise and answer questions.
   1. People use models and simulations to generate new understanding and knowledge.
   2. Models use different levels of abstraction to represent phenomena.
   3. Hypotheses can be formulated, refined, and tested using models and simulations.
   4. Simulations can facilitate extensive and rapid testing of models.

III. **Data**: Data and information facilitate the creation of knowledge.

A. People use computer programs to process information to gain insight and knowledge.
   1. Computers can be used to find patterns in, and test hypotheses about, digitally represented information.
   2. Insight and knowledge can result from translating and transforming digitally represented information.

B. Computing facilitates exploration and the discovery of connections in information.
   1. *Big Data* (use of large datasets) provides new opportunities and new challenges for extracting information and knowledge.
   2. Scalability, of systems and analytical approaches, is an important consideration when datasets are large.
   3. Metadata can increase the effective use of data or a dataset by providing additional information about various aspects of that data.

C. Computational manipulation of information requires consideration of representation, storage, security, and transmission.
   1. There are trade-offs involved in the many possible ways to represent digital and non-digital information as digital data.
   2. Data is stored in many formats depending on its characteristics—such as size and intended use—so that it can be manipulated computationally.

IV. **Algorithms**: Algorithms are used to develop and express solutions to computational problems.

A. An algorithm is a precise sequence of instructions for a process that can be executed by a computer.
   1. Sequencing, selection, iteration, and recursion are building blocks of algorithms.
   2. Algorithms can be combined to make new algorithms.
   3. Different algorithms can be developed to solve the same problem.

B. Algorithms are expressed using languages.
   1. Languages for algorithms include natural language, pseudo-code, and visual and textual programming languages.
   2. The language used to express an algorithm can be different from the programming language used to implement the algorithm.
   3. Different languages are better suited for expressing different algorithms.
   4. The language used to express an algorithm can affect characteristics such as clarity or readability, but not whether an algorithmic solution exists.

C. Algorithms can solve many, but not all, problems.
   1. Many problems can be solved in a reasonable time.
   2. Some problems can be solved, but heuristic approaches are necessary to solve them in a reasonable time.
   3. Some problems cannot be solved using any algorithm.

D. Algorithms are evaluated analytically and empirically.
   1. Algorithms can be evaluated using many criteria (e.g., efficiency, correctness, and clarity).
   2. Different correct algorithms for the same problem can have different efficiencies.

V. **Programming**: Programming enables problem solving, human expression, and creation of knowledge.

A. Programs are written to execute algorithms.
   1. Programming requires an understanding of how instructions are processed.
   2. Programs are executed to automate processes.
   3. A single program can be run multiple times and on many machines.
   4. Executable programs increase the scale of problems that can be addressed.

B. Programming is facilitated by appropriate abstractions.
   1. Functions are re-usable programming abstractions.
   2. Parameterization can be used to generalize a specific solution.
   3. Data abstraction provides a means of separating behavior from implementation.
   4. Application Program Interfaces (APIs) and libraries simplify complex programming tasks.

C. Programs are developed and used by people.
   1. Programs are developed to solve problems.
   2. Developing programs is an iterative process.
   3. Finding and eliminating errors is an essential part of developing programs.
   4. Documentation is a necessary part of developing maintainable programs.
   5. Programs are evaluated for their correctness and style.

D. Programming uses mathematical and logical concepts.
   1. Programming uses numerical concepts including real numbers and integers.
   2. Programming uses applications of logical concepts including Boolean algebra.
   3. Sets and collections are tools for solving computational problems.

VI. **Internet**: The Internet pervades modern computing.

A. The Internet is a network of autonomous systems.
   1. The Internet connects devices and networks all over the world.
   2. The Internet and the systems built on it facilitate collaboration.
   3. The Internet is built on evolving standards including those for addresses and names.

B. Characteristics of the Internet and the systems built on it influence their use.
   1. Hierarchy and redundancy help systems scale.
   2. Interfaces and protocols enable widespread use.
   3. The size and speed of systems affect their use.

C. Cybersecurity is an important concern for the Internet and systems built on it.
   1. The trust model of the Internet involves tradeoffs.
   2. Cryptography is essential to many models of cybersecurity.
   3. Implementing cybersecurity has software, hardware, and human components.

VII. **Impact**: Computing has global impacts.

A. Computing affects communication, interaction, and cognition.
   1. Computing enhances communication, fostering new ways to communicate and collaborate.
   2. Widespread access to information facilitates identification of problems, development of solutions, and dissemination of results.
   3. Computing enhances human capabilities (e.g., through the use of cyber-physical systems and assistive technologies).
   4. The Internet and the web have a profound impact on society.

B. Computing enables innovation in nearly every field.
   1. Computational approaches and data analysis enable innovation.
   2. Computing enables innovation by providing access to, and sharing of, information.

C. Computing has both beneficial and harmful effects.
   1. Innovations enabled by computing raise legal and ethical concerns.
   2. Privacy and security concerns arise in the development and use of computational systems and artifacts.
   3. Technology enables collection, use, and exploitation of information about, by, and for individuals, groups, and institutions.
   4. Widespread access to digitized information raises questions about intellectual property.

D. Computing is situated within economic, social, and cultural contexts.
   1. Computing innovations both influence and are influenced by the contexts in which they are designed and the contexts in which they are used.
   2. The global distribution of computing resources raises issues of equity, access, and power.