CIS Standards

(Updated November 2016)

Computer Science Principles

for all Programs of Study

Please select the Computer Science Principles standards that match the program of study to make sure you have the correct standard reference number.

Computer Science Principles

Course Description:
Computer Science Principles introduces students to the foundations of computer science with a focus on how computing powers the world. Along with the fundamentals of computing, you will learn to analyze data, create technology that has a practical impact, and gain a broader understanding of how computer science impacts people and society.

Course Code: 270703

Endorsements to Teach:
IT, Math

Programs of Study to which this Course applies:
Information Technology, Computer Science, Software Development

CIS. HS. 7. 21

Recognize and define computational problems.

CIS. HS. 7. 21. e Use user-centered research and design techniques (e.g., surveys, interviews) to create innovative software solutions.

CIS. HS. 7. 21. f Describe algorithms using foundational principles of sequence, selection, and iteration with non-code techniques (e.g., pseudo code, flow charts, sequence diagrams).

CIS. HS. 7. 21. g Identify and discuss patterns between previously solved computational problems and new problem scenarios.

CIS. HS. 7. 21. h Explain the difference between algorithms that run in a reasonable amount of time, those that do not run in a reasonable amount of time, and those that are not solvable with a computer.

CIS. HS. 7. 21. i Describe one or more problems that are difficult to solve using computers and explain why it is hard.

CIS. HS. 7. 21. j Describe linear and binary search techniques and explain their appropriateness for a given data set.

CIS. HS. 7. 21. k Use empirical data to analyze multiple sorting techniques and explain why one might outperform another in a given scenario.

CIS. HS. 7. 22

Develop and use abstractions in computational artifacts.

CIS. HS. 7. 22. f Deconstruct a complex problem into distinct functional parts using predefined constructs of a programming language (e.g., functions, procedures, methods).

CIS. HS. 7. 22. g Develop higher-order functions that use parameters to specialize behaviors.

CIS. HS. 7. 22. h Create and use an abstraction of data in order to manage problem complexity (e.g., using a list instead of multiple discrete variables).

CIS. HS. 7. 22. i Provide a qualitative explanation for the advantage of a given data abstraction over another technique.

CIS. HS. 7. 22. j Convert numbers between binary, decimal, and hexadecimal.

CIS. HS. 7. 22. k Explain how abstractions on binary numbers are used to represent and store various kinds of data in computer systems (e.g., hexadecimal color codes, Unicode characters, musical pitches, videos).

CIS. HS. 7. 22. l Analyze the tradeoffs among various representations of a type of digital information (e.g., lossy versus lossless compression, encrypted vs. unencrypted, various image representations).

CIS. HS. 7. 23

Create computational artifacts.

CIS. HS. 7. 23. e Develop programs that use nested loops and multiple or nested branches.

CIS. HS. 7. 23. f Implement interactive programs that process use input and/or respond to events in the system.

CIS. HS. 7. 23. g Store data in multiple variables, and lists/arrays based on user input and program specifications.

CIS. HS. 7. 23. h Develop a computational artifact that coordinates behavior across multiple units (e.g., multiple code files, code attached to multiple sprites or app screens).

CIS. HS. 7. 23. i Construct a computational artifact that effectively uses at least one form of persistent data storage (e.g., reading/writing text files, storing to local databases).

CIS. HS. 7. 23. j Integrate grade-level appropriate mathematical techniques, concepts, and processes in the creation of computing artifacts.

CIS. HS. 7. 24

Use data to understand and model real-world situations.

CIS. HS. 7. 24. f Discuss techniques used to store, process, and retrieve different amounts of information (e.g., files, databases, data warehouses).

CIS. HS. 7. 24. g Apply basic techniques for locating and collecting small- and large-scale data sets (e.g., creating and distributing user surveys, accessing real-world data sets).

CIS. HS. 7. 24. h Explain the processes used to collect, transform, and analyze data to solve a problem using computational tools (e.g., use an app or spreadsheet form to collect data, decide which data to use or ignore, and choose a visualization method.).

CIS. HS. 7. 24. i Explain how modeling and simulation can be used to explore natural phenomena using a concrete example (e.g., flocking behaviors, queueing, life cycles).

Computer Science Principles

Course Description:

Computer Science Principles introduces students to the foundations of computer science with a focus on how computing powers the world. Along with the fundamentals of computing, you will learn to analyze data, create technology that has a practical impact, and gain a broader understanding of how computer science impacts people and society.

Course Code: 270703

Endorsements to Teach:

IT, Math

Programs of Study to which this Course applies:

Information Technology, Computer Science, Software Development

CIS. HS. 8. 1

Recognize and define computational problems.

CIS. HS. 8. 1. e Use user-centered research and design techniques (e.g., surveys, interviews) to create innovative software solutions.

CIS. HS. 8. 1. f Describe algorithms using foundational principles of sequence, selection, and iteration with non-code techniques (e.g., pseudo code, flow charts, sequence diagrams).

CIS. HS. 8. 1. g Identify and discuss patterns between previously solved computational problems and new problem scenarios.

CIS. HS. 8. 1. h Explain the difference between algorithms that run in a reasonable amount of time, those that do not run in a reasonable amount of time, and those that are not solvable with a computer.

CIS. HS. 8. 1. i Describe one or more problems that are difficult to solve using computers and explain why it is hard.

CIS. HS. 8. 1. j Describe linear and binary search techniques and explain their appropriateness for a given data set.

CIS. HS. 8. 1. k Use empirical data to analyze multiple sorting techniques and explain why one might outperform another in a given scenario.

CIS. HS. 8. 2

Develop and use abstractions in computational artifacts.

CIS. HS. 8. 2. f Deconstruct a complex problem into distinct functional parts using predefined constructs of a programming language (e.g., functions, procedures, methods).

CIS. HS. 8. 2. g Develop higher-order functions that use parameters to specialize behaviors.

CIS. HS. 8. 2. h Create and use an abstraction of data in order to manage problem complexity (e.g., using a list instead of multiple discrete variables).

CIS. HS. 8. 2. i Provide a qualitative explanation for the advantage of a given data abstraction over another technique.

CIS. HS. 8. 2. j Convert numbers between binary, decimal, and hexadecimal.

CIS. HS. 8. 2. k Explain how abstractions on binary numbers are used to represent and store various kinds of data in computer systems (e.g., hexadecimal color codes, Unicode characters, musical pitches, videos).

CIS. HS. 8. 2. l Analyze the tradeoffs among various representations of a type of digital information (e.g., lossy vs. lossless compression, encrypted vs. unencrypted, various image representations).

CIS. HS. 8. 3

Create computational artifacts.

CIS. HS. 8. 3. e Develop programs that use nested loops and multiple or nested branches.

CIS. HS. 8. 3. f Implement interactive programs that process use input and/or respond to events in the system.

CIS. HS. 8. 3. g Store data in multiple variables, and lists/arrays based on user input and program specifications.

CIS. HS. 8. 3. h Develop a computational artifact that coordinates behavior across multiple units (e.g., multiple code files, code attached to multiple sprites or app screens).

CIS. HS. 8. 3. i Construct a computational artifact that effectively uses at least one form of persistent data storage (e.g., reading/writing text files, storing to local databases).

CIS. HS. 8. 3. j Integrate grade-level appropriate mathematical techniques, concepts, and processes in the creation of computing artifacts.

CIS. HS. 8. 4

Use data to understand and model real-world situations.

CIS. HS. 8. 4. f Discuss techniques used to store, process, and retrieve different amounts of information (e.g., files, databases, data warehouses).

CIS. HS. 8. 4. g Apply basic techniques for locating and collecting small- and large-scale data sets (e.g., creating and distributing user surveys, accessing real-world data sets).

Computer Science Principles

Course Description:

Computer Science Principles introduces students to the foundations of computer science with a focus on how computing powers the world. Along with the fundamentals of computing, you will learn to analyze data, create technology that has a practical impact, and gain a broader understanding of how computer science impacts people and society.

Course Code: 270703

Endorsements to Teach:
IT, Math

Programs of Study to which this Course applies:
Information Technology, Computer Science, Software Development

CIS. HS. 9. 1

Recognize and define computational problems.

CIS. HS. 9. 1. e Use user-centered research and design techniques (e.g., surveys, interviews) to create innovative software solutions.

CIS. HS. 9. 1. f Describe algorithms using foundational principles of sequence, selection, and iteration with non-code techniques (e.g., pseudo code, flow charts, sequence diagrams).

CIS. HS. 9. 1. g Identify and discuss patterns between previously solved computational problems and new problem scenarios.

CIS. HS. 9. 1. h Explain the difference between algorithms that run in a reasonable amount of time, those that do not run in a reasonable amount of time, and those that are not solvable with a computer.

CIS. HS. 9. 1. i Describe one or more problems that are difficult to solve using computers and explain why it is hard.

CIS. HS. 9. 1. j Describe linear and binary search techniques and explain their appropriateness for a given data set.

CIS. HS. 9. 1. k Use empirical data to analyze multiple sorting techniques and explain why one might outperform another in a given scenario.

CIS. HS. 9. 2

Develop and use abstractions in computational artifacts.

CIS. HS. 9. 2. f Deconstruct a complex problem into distinct functional parts using predefined constructs of a programming language (e.g., functions, procedures, methods).

CIS. HS. 9. 2. g Develop higher-order functions that use parameters to specialize behaviors.

CIS. HS. 9. 2. h Create and use an abstraction of data in order to manage problem complexity (e.g., using a list instead of multiple discrete variables).

CIS. HS. 9. 2. i Provide a qualitative explanation for the advantage of a given data abstraction over another technique.

CIS. HS. 9. 2. j Convert numbers between binary, decimal, and hexadecimal.

CIS. HS. 9. 2. k Explain how abstractions on binary numbers are used to represent and store various kinds of data in computer systems (e.g., hexadecimal color codes, Unicode characters, musical pitches, videos).

CIS. HS. 9. 2. l Analyze the tradeoffs among various representations of a type of digital information (e.g., lossy vs. lossless compression, encrypted vs. unencrypted, various image representations).

CIS. HS. 9. 3

Create computational artifacts.

CIS. HS. 9. 3. e Develop programs that use nested loops and multiple or nested branches.

CIS. HS. 9. 3. f Implement interactive programs that process use input and/or respond to events in the system.

CIS. HS. 9. 3. g Store data in multiple variables, and lists/arrays based on user input and program specifications.

CIS. HS. 9. 3. h Develop a computational artifact that coordinates behavior across multiple units (e.g., multiple code files, code attached to multiple sprites or app screens).

CIS. HS. 9. 3. i Construct a computational artifact that effectively uses at least one form of persistent data storage (e.g., reading/writing text files, storing to local databases).

CIS. HS. 9. 3. j Integrate grade-level appropriate mathematical techniques, concepts, and processes in the creation of computing artifacts.

CIS. HS. 9. 4

Use data to understand and model real-world situations.

CIS. HS. 9. 4. f Discuss techniques used to store, process, and retrieve different amounts of information (e.g., files, databases, data warehouses).

CIS. HS. 9. 4. g Apply basic techniques for locating and collecting small- and large-scale data sets (e.g., creating and distributing user surveys, accessing real-world data sets).