Computer Science A (CSA)

Lesson 1.1

The Computer Science Teachers Association Standards

CSTA.3B.CD1 Computer Science Principles (CP)

Discuss the impact of modifications on the functionality of application programs.

CSTA.3B.CT10 Computer Science Principles (CP)

Decompose a problem by defining new functions and classes.

Computer Science A (CSA)

Lesson 1.2

Next Generation Science Standards

Science and Engineering Practice - Using Mathematics and Computational Thinking

> Use simple limit cases to test mathematical expressions, computer programs, algorithms, or simulations of a process or system to see if a model "makes sense" by comparing the outcomes with what is known about the real world.

Computer Science A (CSA)

Lesson 1.2

The Computer Science Teachers Association Standards

CSTA.3B.CD1 Computer Science Principles (CP)

Discuss the impact of modifications on the functionality of application programs.

CSTA.3B.CT10 Computer Science Principles (CP)

Decompose a problem by defining new functions and classes.

Computer Science A (CSA)

Lesson 2.1

Next Generation Science Standards

HS.ETS1.3 - Engineering Design

Evaluate a solution to a complex, real-world problem based on prioritized criteria and tradeoffs that account for a range of constraints, including cost, safety, reliability, and aesthetics, as well as possible social, cultural, and environmental impacts.

Computer Science A (CSA)

Lesson 2.1

The Computer Science Teachers Association Standards

CSTA.3B.CD1 Computer Science Principles (CP)

Discuss the impact of modifications on the functionality of application programs.

CSTA.3B.CI1 Computer Science Principles (CP)

Demonstrate ethical use of modern communication media and devices.

CSTA.3B.CT10 Computer Science Principles (CP)

Decompose a problem by defining new functions and classes.

Computer Science A (CSA)

Lesson 2.2

The Computer Science Teachers Association Standards

CSTA.3B.CD1 Computer Science Principles (CP)

Discuss the impact of modifications on the functionality of application programs.

CSTA.3B.CI1 Computer Science Principles (CP)

Demonstrate ethical use of modern communication media and devices.

CSTA.3B.CT10 Computer Science Principles (CP)

Decompose a problem by defining new functions and classes.

Computer Science A (CSA)

Lesson 2.3

Next Generation Science Standards

HS.ETS1.2 - Engineering Design

Design a solution to a complex, real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.

Science and Engineering Practice - Using Mathematics and Computational Thinking

Use simple limit cases to test mathematical expressions, computer programs, algorithms, or simulations of a process or system to see if a model "makes sense" by comparing the outcomes with what is known about the real world.

Science and Engineering Practice - Constructing Explanations and Designing Solutions

Design, evaluate, and/or refine a solution to a complex, real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.

Computer Science A (CSA)

Lesson 2.3

The Computer Science Teachers Association Standards

CSTA.3B.CD1 Computer Science Principles (CP)

Discuss the impact of modifications on the functionality of application programs.

CSTA.3B.CD3 Computer Science Principles (CP)

Identify and select the most appropriate file format based on tradeoffs (e.g., accuracy, speed, ease of manipulation).

CSTA.3B.CL1 Computer Science Principles (CP)

Use project collaboration tools, version control systems, and Integrated Development Environments (IDEs) while working on a collaborative software project.

CSTA.3B.CPP2 Computer Science Principles (CP)

Use tools of abstraction to decompose a large-scale computational problem (e.g., procedural abstraction, object-oriented design, functional design).

CSTA.3B.CT10 Computer Science Principles (CP)

Decompose a problem by defining new functions and classes.

Computer Science A (CSA)

Lesson 3.1

The Computer Science Teachers Association Standards

CSTA.3B.CD1 Computer Science Principles (CP)

Discuss the impact of modifications on the functionality of application programs.

CSTA.3B.CI1 Computer Science Principles (CP)

Demonstrate ethical use of modern communication media and devices.

CSTA.3B.CI2 Computer Science Principles (CP)

Analyze the beneficial and harmful effects of computing innovations.

CSTA.3B.CPP2 Computer Science Principles (CP)

Use tools of abstraction to decompose a large-scale computational problem (e.g., procedural abstraction, object-oriented design, functional design).

CSTA.3B.CPP5 Computer Science Principles (CP)

Deploy principles of security by implementing encryption and authentication strategies.

CSTA.3B.CT3 Computer Science Principles (CP)

Critically examine classical algorithms and implement an original algorithm.

CSTA.3B.CT4 Computer Science Principles (CP)

Evaluate algorithms by their efficiency, correctness, and clarity.

CSTA.3B.CT10 Computer Science Principles (CP)

Decompose a problem by defining new functions and classes.

CSTA.3B.CT11 Computer Science Principles (CP)

Demonstrate concurrency by separating processes into threads and dividing data into parallel streams.

Computer Science A (CSA)

Lesson 3.2

The Computer Science Teachers Association Standards

CSTA.3B.CD1 Computer Science Principles (CP)

Discuss the impact of modifications on the functionality of application programs.

CSTA.3B.CI1 Computer Science Principles (CP)

Demonstrate ethical use of modern communication media and devices.

CSTA.3B.CI2 Computer Science Principles (CP)

Analyze the beneficial and harmful effects of computing innovations.

CSTA.3B.CPP2 Computer Science Principles (CP)

Use tools of abstraction to decompose a large-scale computational problem (e.g., procedural abstraction, object-oriented design, functional design).

CSTA.3B.CT10 Computer Science Principles (CP)

Decompose a problem by defining new functions and classes.

Computer Science A (CSA)

Lesson 3.3

The Computer Science Teachers Association Standards

CSTA.3B.CD1 Computer Science Principles (CP)

Discuss the impact of modifications on the functionality of application programs.

CSTA.3B.CI1 Computer Science Principles (CP)

Demonstrate ethical use of modern communication media and devices.

CSTA.3B.CI2 Computer Science Principles (CP)

Analyze the beneficial and harmful effects of computing innovations.

CSTA.3B.CPP2 Computer Science Principles (CP)

Use tools of abstraction to decompose a large-scale computational problem (e.g., procedural abstraction, object-oriented design, functional design).

CSTA.3B.CT10 Computer Science Principles (CP)

Decompose a problem by defining new functions and classes.

Computer Science A (CSA)

Lesson 3.4

Next Generation Science Standards

HS.ETS1.2 - Engineering Design

> Design a solution to a complex, real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.

DCI - ETS1.C - Engineering Design - Optimizing the Design Solution

> Criteria may need to be broken down into simpler ones that can be approached systematically, and decisions about the priority of certain criteria over others (tradeoffs) may be needed. (secondary to HS-PS1-6)

Science and Engineering Practice - Using Mathematics and Computational Thinking

> Use simple limit cases to test mathematical expressions, computer programs, algorithms, or simulations of a process or system to see if a model “makes sense” by comparing the outcomes with what is known about the real world.

Science and Engineering Practice - Constructing Explanations and Designing Solutions

> Design, evaluate, and/or refine a solution to a complex, real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.

Science and Engineering Practice - Obtaining, Evaluating, and Communicating Information

> Gather, read, and evaluate scientific and/or technical information from multiple authoritative sources, assessing the evidence and usefulness of each source.

Computer Science A (CSA)

Lesson 3.4

The Computer Science Teachers Association Standards

CSTA.3B.CD1 Computer Science Principles (CP)

Discuss the impact of modifications on the functionality of application programs.

CSTA.3B.CD3 Computer Science Principles (CP)

Identify and select the most appropriate file format based on tradeoffs (e.g., accuracy, speed, ease of manipulation).

CSTA.3B.CI2 Computer Science Principles (CP)

Analyze the beneficial and harmful effects of computing innovations.

CSTA.3B.CL1 Computer Science Principles (CP)

Use project collaboration tools, version control systems, and Integrated Development Environments (IDEs) while working on a collaborative software project.

CSTA.3B.CPP2 Computer Science Principles (CP)

Use tools of abstraction to decompose a large-scale computational problem (e.g., procedural abstraction, object-oriented design, functional design).

CSTA.3B.CT4 Computer Science Principles (CP)

Evaluate algorithms by their efficiency, correctness, and clarity.

CSTA.3B.CT10 Computer Science Principles (CP)

Decompose a problem by defining new functions and classes.

Computer Science A (CSA)

Lesson 4.1

The Computer Science Teachers Association Standards

CSTA.3B.CD1 Computer Science Principles (CP)

Discuss the impact of modifications on the functionality of application programs.

CSTA.3B.CI2 Computer Science Principles (CP)

Analyze the beneficial and harmful effects of computing innovations.

CSTA.3B.CPP2 Computer Science Principles (CP)

Use tools of abstraction to decompose a large-scale computational problem (e.g., procedural abstraction, object-oriented design, functional design).

CSTA.3B.CT10 Computer Science Principles (CP)

Decompose a problem by defining new functions and classes.

CSTA.3B.CT6 Computer Science Principles (CP)

Compare and contrast simple data structures and their uses (e.g., arrays and lists).

Computer Science A (CSA)

Lesson 4.2

Next Generation Science Standards

Science and Engineering Practice - Using Mathematics and Computational Thinking

Apply techniques of algebra and functions to represent and solve scientific and engineering problems.

Computer Science A (CSA)

Lesson 4.2

Common Core State Standards for Mathematics

A.CED.3 - Creating Equations

Represent constraints by equations or inequalities, and by systems of equations and/or inequalities, and interpret solutions as viable or non-viable options in a modeling context. For example, represent inequalities describing nutritional and cost constraints on combinations of different foods.

A.REI.6 - Reasoning with Equations and Inequalities

Solve systems of linear equations exactly and approximately (e.g., with graphs), focusing on pairs of linear equations in two variables.

A.REI.12 - Reasoning with Equations and Inequalities

Graph the solutions to a linear inequality in two variables as a half-plane (excluding the boundary, in the case of a strict inequality), and graph the solution set to a system of linear inequalities in two variables as the intersection of the corresponding half-planes.

Computer Science A (CSA)

Lesson 4.2

The Computer Science Teachers Association Standards

CSTA.3B.CD1 Computer Science Principles (CP)

Discuss the impact of modifications on the functionality of application programs.

CSTA.3B.CD3 Computer Science Principles (CP)

Identify and select the most appropriate file format based on tradeoffs (e.g., accuracy, speed, ease of manipulation).

CSTA.3B.CPP2 Computer Science Principles (CP)

Use tools of abstraction to decompose a large-scale computational problem (e.g., procedural abstraction, object-oriented design, functional design).

CSTA.3B.CT10 Computer Science Principles (CP)

Decompose a problem by defining new functions and classes.

Computer Science A (CSA)

Lesson 4.2

The Computer Science Teachers Association Standards

CSTA.3B.CD1 Computer Science Principles (CP)

Discuss the impact of modifications on the functionality of application programs.

CSTA.3B.CD3 Computer Science Principles (CP)

Identify and select the most appropriate file format based on tradeoffs (e.g., accuracy, speed, ease of manipulation).

CSTA.3B.CPP2 Computer Science Principles (CP)

Use tools of abstraction to decompose a large-scale computational problem (e.g., procedural abstraction, object-oriented design, functional design).

CSTA.3B.CT10 Computer Science Principles (CP)

Decompose a problem by defining new functions and classes.

Computer Science A (CSA)

Lesson 4.3

Common Core State Standards for English Language Arts 11-12th Grade

11-12.W.2.d - Writing

Use precise language, domain-specific vocabulary, and techniques, such as metaphor, simile, and analogy, to manage the complexity of the topic.

11-12.RST.7 - Reading Science/Technical

Integrate and evaluate multiple sources of information presented in diverse formats and media (e.g., quantitative data, video, multimedia) in order to address a question or solve a problem.

Computer Science A (CSA)

Lesson 4.3

Next Generation Science Standards

HS.ETS1.2 - Engineering Design

Design a solution to a complex, real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.

DCI - ETS1.A - Engineering Design - Defining and Delimiting Engineering Problems

Criteria and constraints also include satisfying any requirements set by society, such as taking issues of risk mitigation into account, and they should be quantified to the extent possible and stated in such a way that one can tell if a given design meets them. (secondary to HS-PS2-3)

DCI - ETS1.B - Engineering Design - Developing Possible Solutions

When evaluating solutions, it is important to take into account a range of constraints, including cost, safety, reliability, and aesthetics, and to consider social, cultural, and environmental impacts. (HS-ETS1-3)

DCI - ETS1.C - Engineering Design - Optimizing the Design Solution

Criteria may need to be broken down into simpler ones that can be approached systematically, and decisions about the priority of certain criteria over others (tradeoffs) may be needed. (secondary to HS-PS1-6)

Science and Engineering Practice - Asking questions and defining problems

Define a design problem that involves the development of a process or system with interacting components and criteria and constraints that may include social, technical, and/or environmental considerations.

Science and Engineering Practice - Planning and Carrying Out Investigations

Plan and conduct an investigation or test a design solution in a safe and ethical manner including considerations of environmental, social, and personal impacts.

Science and Engineering Practice - Using Mathematics and Computational Thinking

Use simple limit cases to test mathematical expressions, computer programs, algorithms, or simulations of a process or system to see if a model “makes sense” by comparing the outcomes with what is known about the real world.

Science and Engineering Practice - Constructing Explanations and Designing Solutions

Design, evaluate, and/or refine a solution to a complex, real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.

Science and Engineering Practice - Engaging in Argument from Evidence

Evaluate competing design solutions to a real-world problem based on scientific ideas and principles, empirical evidence, and/or logical arguments regarding relevant factors (e.g., economic, societal, environmental, ethical considerations).

Science and Engineering Practice - Obtaining, Evaluating, and Communicating Information

Gather, read, and evaluate scientific and/or technical information from multiple authoritative sources, assessing the evidence and usefulness of each source.

Science and Engineering Practice - Obtaining, Evaluating, and Communicating Information

Evaluate the validity and reliability of and/or synthesize multiple claims, methods, and/or designs that appear in scientific and technical texts or media reports, verifying the data when possible. Communicate scientific and/or technical information or ideas (e.g., about phenomena and/or the process of development and the design and performance of a proposed process or system) in multiple formats (i.e., orally, graphically, textually, mathematically).

Computer Science A (CSA)

Lesson 4.3

Common Core State Standards for English Language Arts 11-12th Grade

11-12.W.2.d - Writing

Use precise language, domain-specific vocabulary, and techniques, such as metaphor, simile, and analogy, to manage the complexity of the topic.

11-12.RST.7 - Reading Science/Technical

Integrate and evaluate multiple sources of information presented in diverse formats and media (e.g., quantitative data, video, multimedia) in order to address a question or solve a problem.

Computer Science A (CSA)

Lesson 4.3

The Computer Science Teachers Association Standards

CSTA.3B.CD1 Computer Science Principles (CP)

Discuss the impact of modifications on the functionality of application programs.

CSTA.3B.CD3 Computer Science Principles (CP)

Identify and select the most appropriate file format based on tradeoffs (e.g., accuracy, speed, ease of manipulation).

CSTA.3B.CI1 Computer Science Principles (CP)

Demonstrate ethical use of modern communication media and devices.

CSTA.3B.CL1 Computer Science Principles (CP)

Use project collaboration tools, version control systems, and Integrated Development Environments (IDEs) while working on a collaborative software project.

CSTA.3B.CL2 Computer Science Principles (CP)

Demonstrate the software life cycle process by participating on a software project team.

CSTA.3B.CL3 Computer Science Principles (CP)

Evaluate programs written by others for readability and usability.

CSTA.3B.CPP2 Computer Science Principles (CP)

Use tools of abstraction to decompose a large-scale computational problem (e.g., procedural abstraction, object-oriented design, functional design).

CSTA.3B.CT10 Computer Science Principles (CP)

Decompose a problem by defining new functions and classes.

CSTA.3B.CT3 Computer Science Principles (CP)

Critically examine classical algorithms and implement an original algorithm.

CSTA.3B.CT4 Computer Science Principles (CP)

Evaluate algorithms by their efficiency, correctness, and clarity.