TRUMBULL PUBLIC SCHOOLS

Trumbull, Connecticut

Advanced Placement Computer Science Principles

Mathematics Department

2018

Curriculum Writing Team

Fran Basbagill	**Department Chair**
Scott Kaminski	**Teacher**
Matthew Scerbo	**Teacher**

Jonathan S. Budd, Ph.D., Assistant Superintendent of Curriculum, Instruction, & Assessments

AP Computer Science Principles
Table of Contents

CORE VALUES AND BELIEFS

The Trumbull High School community engages in an environment conducive to learning which believes that all students will **read and write effectively**, therefore communicating in an articulate and coherent manner. All students will participate in activities **that present problem-solving through critical thinking**. Students will use technology as a tool applying it to decision making. We believe that by fostering self-confidence, self-directed and student-centered activities, we will promote **independent thinkers and learners**. We believe **ethical conduct** to be paramount in sustaining the welcoming school climate that we presently enjoy.

Approved 8/26/2011

INTRODUCTION & PHILOSOPHY

AP Computer Science Principles utilizes students' creative, problem-solving skills to design and program mobile applications using App Inventor for Android. App Inventor is a visual language that enables novice programmers to create powerful mobile applications that interact with the web and with other phones. Students will learn how to access the world of mobile services and applications as creators, not just consumers. Students will learn to create useful apps for real-world contexts, as they develop programming and problem-solving skills. Students will also explore the exciting world of computer science from the perspective of mobile computing and its increasingly important effects on society. The course follows a Mobile Computer Science Principles (CSP) curriculum based on the theme of mobile computing and using the visual programming language MIT App Inventor.

COURSE GOALS

AP Computer Science Principles satisfies three sets of goals:

- The Learning Objectives of the 2017 College Board Curriculum Framework for Advanced Placement Computer Science Principles;
- The Standards of the 2017 Computer Science Teachers' Association (CSTA); and
- The Standards of the 2016 International Society for Technology in Education Standards.

Each Unit presents the Goals appropriate for that Unit.

COURSE ENDURING UNDERSTANDINGS

Students will understand that . . .

- developments in computing have far-reaching effects on society and have led to significant innovations. The developments have implications for individuals, society, commercial markets, and innovation.
- computing is a creative discipline in which creation takes many forms, such as remixing digital music, generating animations, developing websites, and writing programs.

- computational thinking requires understanding and applying abstraction at multiple levels, such as privacy in social networking applications, logic gates and bits, and the human genome project.
- the results and artifacts of computation and the computational techniques and strategies that generate them can be understood intrinsically both for what they are as well as for what they produce. They can also be analyzed and evaluated by applying aesthetic, mathematical, pragmatic, and other criteria.
- communicating effectively with respect to computation and the impact of technology is essential. This is true on both the local and global levels. Good communication leads to productive collaboration.
- innovation can occur when people work together or independently. People working collaboratively can often achieve more than individuals working alone. Learning to collaborate effectively includes drawing on diverse perspectives, skills, and the backgrounds of peers to address complex and open-ended problems.

COURSE ESSENTIAL QUESTIONS

- How can a creative development process affect the creation of computational artifacts?
- How can computing and the use of computational tools foster creative expression?
- How can computing extend traditional forms of human expression and experience?
- How are vastly different kinds of data, physical phenomena, and mathematical concepts represented on a computer?
- How does abstraction help us in writing programs, creating computational artifacts, and solving problems?
- How can computational models and simulations help generate new understanding and knowledge?
- How can computation be employed to help people process data and information to gain insight and knowledge?
- How can computation be employed to facilitate exploration and discovery when working with data?
- What considerations and trade-offs arise in the computational manipulation of data?
- What opportunities do large data sets provide for solving problems and creating knowledge?
- How are algorithms implemented and executed on computers and computational devices?
- Why are some languages better than others when used to implement algorithms?
- What kinds of problems are easy, what kinds are difficult, and what kinds are impossible to solve algorithmically?
- How are algorithms evaluated?
- How are programs developed to help people, organizations, or society solve problems?
- How are programs used for creative expression, to satisfy personal curiosity, or to create new knowledge?
- How do computer programs implement algorithms?
- How does abstraction make the development of computer programs possible?
- How do people develop and test computer programs?
- Which mathematical and logical concepts are fundamental to computer programming?
- What is the Internet? How is it built? How does it function?
- What aspects of the Internet's design and development have helped it scale and flourish?
- How is cybersecurity impacting the ever-increasing number of Internet users?

- How does computing enhance human communication, interaction, and cognition?
- How does computing enable innovation?
- What are some potential beneficial and harmful effects of computing?
- How do economic, social, and cultural contexts influence innovation and the use of computing?

COURSE KNOWLEDGE & SKILLS

Students will know . . .

- how to draw connections between different computing concepts.
- how to engage in the creative aspects of computing by designing and developing interesting computational artifacts as well as by applying computing techniques to creatively solve problems.
- how to use abstraction to develop models and simulations of natural and artificial phenomena, use them to make predictions about the world, and analyze their efficacy and validity.
- how to design and produce solutions, models, and artifacts, and evaluate and analyze their own computational work as well as the computational work of others have produced.
- how to describe computation and the impact of technology and computation, 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.
- how to collaborate on a number of activities, including the investigation of questions using data sets and the production of computational artifacts.

Students will be able to . . .

- identify impacts of computing.
- describe connections between people and computing.
- explain connections between computing concepts.
- create a computational artifact with a practical, personal, or societal intent.
- select appropriate techniques to develop a computational artifact.
- use appropriate algorithmic and information management principles.
- explain how data, information, or knowledge is represented for computational use.
- explain how abstractions are used in computation or modeling.
- describe modeling in a computational context.
- evaluate a proposed solution to a problem.
- locate and correct errors.
- explain how an artifact functions.
- justify appropriateness and correctness of a solution, model, or artifact.
- explain the meaning of a result in context.
- describe computation with accurate and precise language, notations, or visualizations.
- summarize the purpose of a computational artifact.
- collaborate with another student in solving a computational problem.
- collaborate with another student in producing an artifact
- share the workload by providing individual contributions to an overall collaborative effort.

- foster a constructive, collaborative climate by resolving conflicts and facilitating the contributions of a partner or team member.
- exchange knowledge and feedback with a partner or team member.
- review and revise their work as needed to create a high-quality artifact.

COURSE SYLLABUS

Course Name

Advanced Placement Computer Science Principles

Level

Advanced Placement

Prerequisites

Successful completion of ACP Geometry

Materials Required

Chromebook or laptop with the Chrome web browser; Android device (phone or tablet) or a laptop with an Android emulator

General Description of the Course

AP Computer Science Principles utilizes students' creative, problem-solving skills to design and program mobile applications using App Inventor for Android. App Inventor is a visual language that enables novice programmers to create powerful mobile applications that interact with the web and with other phones. Students will learn how to access the world of mobile services and applications as creators, not just consumers. Students will learn to create useful apps for real-world contexts, as they develop programming and problem-solving skills. Students will also explore the exciting world of computer science from the perspective of mobile computing and its increasingly important effect on society. If students plan on taking both AP Computer Science classes offered at THS, it is recommended that AP Computer Science be is taken before AP Computer Science A.

Assured Assessments

Formative Assessments:
Formative assessments can include, but are not limited to:

- Programming tasks on Hour of Code website (Unit 1)
- Creation and use of ePortfolios (Units 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13)
- Discussions related to technology and computer science (Units 1, 2, 3, 4, 7, 8, 10)
- Creation of and writing about apps (Units 2, 3, 4, 5, 7, 8, 10, 11, 13)
- Practice questions and released items related to College Board Examination (Unit 12)

Summative Assessments:

- Common end-of-unit assessment (Units 2, 3, 4, 7, 8, 10)
- Programming performance task (Units 5, 11)
- Impact of computing innovation performance task (Units 6, 9)
- Full released College Board Examination (Unit 12)

Core Resources

- Abelson, Hal, Ken Ledeen, and Harry Lewis. "Blown to Bits: Your Life, Liberty, and Happiness after the Digital Explosion." http://www.bitsbook.com/. Web.
- MIT App Inventor. ai2.appinventor.mit.edu. Web.
- "Mobile CSP." https://mobilecsp-2017.appspot.com/mobilecsp/course. Web.

UNIT 1
Getting Started: Preview and Setup

Unit 1 provides a brief overview of the course, emphasizing its main theme: learning the principles of computer science while building socially useful mobile apps. The hands-on work focuses on setting up the students' environment, including their programming environment and online portfolios. Students are provided a brief introduction to blocks-based programming and are also given a brief introduction to *Blown to Bits*, which will be used, along with current events and news articles, as a resource throughout the course to learn about the impact of computer science on society.

Unit Goals

At the completion of this unit, students will:

The following Unit Goals align with the 2017 College Board Curriculum Framework for Advanced Placement Computer Science Principles.

1.1.1	Apply a creative development process when creating computational artifacts.
1.2.1	Create a computational artifact for creative expression.
4.1.1	Develop an algorithm for implementation in a program.
4.1.2	Express an algorithm in a language.
4.2.4	Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity.
5.1.2	Develop a correct program to solve problems.
5.1.3	Collaborate to develop a program.
5.2.1	Explain how programs implement algorithms.

The following Unit Goals align with the 2017 Computer Science Teachers' Association (CSTA) Computer Science Standards.

3B-AP-11	Evaluate algorithms in terms of their efficiency, correctness, and clarity.
3A-AP-15	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.

3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
3B-AP-13	Illustrate the flow of execution of a recursive algorithm.

The following Unit Goal aligns with the 2016 International Society for Technology in Education Standards.

ISTE Computational Thinker (Standard 5)	Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.

Unit Essential Questions

- What is graphical blocks-based programming?
- How will we create apps for our mobile devices?
- Why is it important to study the impact of computer technology?

Scope and Sequence

1. Overview of course structure and requirements
2. Sample programming tasks, using Blockly and Hour of Code
3. Creating a Google site to be used as an aPortfolio
4. Setting up Chromebooks and mobile devices to use App Inventor
5. Introduction of the online textbook *Blown to Bits: Your Life, Liberty, and Happiness after the Digital Explosion*

Assured Assessments

Formative Assessment:
Students will participate in solving various programming tasks on the Hour of Code website, and will also create a Google site to be used as their ePortfolio. They will also engage in various discussions about technology in general. These tasks will assess students' understanding of essential concepts and skills.

Summative Assessment:
There is no summative assessment for this first unit, as it sets up the rest of the year for the course.

Resources

Core

- Abelson, Hal, Ken Ledeen, and Harry Lewis. "Blown to Bits: Your Life, Liberty, and Happiness after the Digital Explosion." http://www.bitsbook.com/. Web.
- Hour of Code. https://hourofcode.com/us/learn. Web.
- MIT App Inventor. ai2.appinventor.mit.edu. Web.
- "Mobile CSP." https://mobilecsp-2017.appspot.com/mobilecsp/course. Web.

Time Allotment

- Approximately 7 school days

UNIT 2
Introduction to Mobile Apps and Pair Programming

Unit 2 focuses on getting to know the App Inventor development environment as well as several key enduring understandings related to abstraction, global impact, and creativity.

Unit Goals

At the completion of this unit, students will:

The following Unit Goals align with the 2017 College Board Curriculum Framework for Advanced Placement Computer Science Principles.

1.1.1	Apply a creative development process when creating computational artifacts.
1.2.1	Create a computational artifact for creative expression.
1.2.3	Create a new computational artifact by combining or modifying existing artifacts.
1.2.4	Collaborate in the creation of computational artifacts.
1.3.1	Use computing tools and techniques for creative expression.
2.1.1	Describe the variety of abstractions used to represent data.
2.1.2	Explain how binary sequences are used to represent digital data.
2.2.3	Identify multiple levels of abstractions that are used when writing programs.
4.1.1	Develop an algorithm for implementation in a program.
4.1.2	Express an algorithm in a language.
5.1.1	Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge.
5.1.2	Develop a correct program to solve problems.
5.2.1	Explain how programs implement algorithms.
5.4.1	Evaluate the correctness of a program.
6.1.1	Explain the abstraction in the Internet and how the Internet functions.
7.1.1	Explain how computing innovations affect communication, interaction, and cognition.
7.1.2	Explain how people participate in a problem-solving process that scales.
7.3.1	Analyze the beneficial and harmful effects of computing.

The following Unit Goals align with the 2017 Computer Science Teachers' Association (CSTA) Computer Science Standards.

3A-CS-01	Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.
3A-CS-02	Compare levels of abstraction and interactions between application software, system software, and hardware layers.
3B-CS-01	Categorize the roles of operating system software.
3B-CS-02	Illustrate ways computing systems implement logic, input, and output through hardware components.
3A-NI-04	Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.
3B-NI-03	Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).
3A-DA-09	Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.
3A-DA-10	Evaluate the tradeoffs in how data elements are organized and where data is stored.
3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
3A-AP-15	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
3A-AP-18	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
3A-AP-19	Systematically design and develop programs for broad audiences by incorporating feedback from users.
3A-AP-20	Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.

3A-AP-21	Evaluate and refine computational artifacts to make them more usable and accessible.
3B-AP-10	Use and adapt classic algorithms to solve computational problems.
3B-AP-11	Evaluate algorithms in terms of their efficiency, correctness, and clarity.
3B-AP-15	Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.
3B-AP-20	Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.
3B-AP-21	Develop and use a series of test cases to verify that a program performs according to its design specifications.
3B-AP-22	Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
3B-AP-23	Evaluate key qualities of a program through a process such as a code review.
3A-IC-24	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
3A-IC-28	Explain the beneficial and harmful effects that intellectual property laws can have on innovation.
3B-IC-25	Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.
3B-IC-26	Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.
3B-IC-27	Predict how computational innovations that have revolutionized aspects of our culture might evolve.
3B-IC-28	Debate laws and regulations that impact the development and use of software.

The following Unit Goals align with the 2016 International Society for Technology in Education Standards.

ISTE Digital Citizen (Standard 2)	Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical.
ISTE Innovative Designer (Standard 4)	Students use a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions.

ISTE Computational Thinker (Standard 5)	Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.
ISTE Creative Communicator (Standard 6)	Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals.

Unit Essential Questions

- How does one use App Inventor and event-driven programming to build a mobile app?
- What are the various hardware and software abstractions that make up a modern digital computer?
- What is the binary number system that underlines all digital representation?

Scope and Sequence

1. Completion of introductory tutorial using images and sound in App Inventor
2. Building on introductory tutorial by adding new features and enhancements
3. Designing own soundboard apps
4. Learning the basic digital structure of computers – binary numbers – via highlighting of the various levels and types of abstraction in computer hardware and software
5. Reading and discussion of Chp. 1 of *Blown to Bits*, focusing on some of the key ethical considerations around the positive and negative impacts of computing

Assured Assessments

Formative Assessment:
Students will answer interactive self-check questions and add to their ePortfolios answers to reflection questions. They will also create two apps, upload them to their ePortfolios, and complete an app write-up. They will also engage in various discussions about technology's impact on society.

Summative Assessment:
Students will take a common end-of-unit assessment scored via a common scoring guide.

Resources

Core

- Abelson, Hal, Ken Ledeen, and Harry Lewis. "Blown to Bits: Your Life, Liberty, and Happiness after the Digital Explosion." http://www.bitsbook.com/. Web.
- Logic.Ly. https://logic.ly/demo/samples. Web.
- MIT App Inventor. ai2.appinventor.mit.edu. Web.
- "Mobile CSP." https://mobilecsp-2017.appspot.com/mobilecsp/course. Web.

Time Allotment

- Approximately 17 school days

UNIT 3
Creating Graphics and Images Bit by Bit

Unit 3 focuses on App Inventor's drawing and painting features to explore concepts related to graphics and images; students also delve deeper into abstraction and begin exploring data and information as they discuss the benefits and costs of image representation techniques.

Unit Goals

At the completion of this unit, students will:

The following Unit Goals align with the 2017 College Board Curriculum Framework for Advanced Placement Computer Science Principles.

1.2.2	Create a computational artifact using computing tools and techniques to solve a problem.
1.2.4	Collaborate in the creation of computational artifacts.
1.3.1	Use computing tools and techniques for creative expression.
2.1.1	Describe the variety of abstractions used to represent data.
2.1.2	Explain how binary sequences are used to represent digital data.
2.2.1	Develop an abstraction when writing a program or creating other computational artifacts.
2.2.2	Use multiple levels of abstraction to write programs.
2.3.1	Use models and simulations to represent phenomena.
3.1.2	Collaborate when processing information to gain insight and knowledge.
3.1.3	Explain the insight and knowledge gained from digitally processed data by using appropriate visualizations, notations, and precise language.
3.2.1	Extract information from data to discover and explain connections or trends.
3.3.1	Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information.
4.1.2	Express an algorithm in a language.
5.1.1	Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge.
5.1.2	Develop a correct program to solve problems.
5.1.3	Collaborate to develop a program.

5.2.1 Explain how programs implement algorithms.

5.3.1 Use abstraction to manage complexity in programs.

5.4.1 Evaluate the correctness of a program.

5.5.1 Employ appropriate mathematical and logical concepts in programming.

7.1.1 Explain how computing innovations affect communication, interaction, and cognition.

7.3.1 Analyze the beneficial and harmful effects of computing.

The following Unit Goals align with the 2017 Computer Science Teachers' Association (CSTA) Computer Science Standards.

3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

3A-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers.

3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

3B-CS-01 Categorize the roles of operating system software.

3B-NI-03 Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).

3A-DA-09 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.

3A-DA-10 Evaluate the tradeoffs in how data elements are organized and where data is stored.

3A-DA-12 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.

3B-DA-07 Evaluate the ability of models and simulations to test and support the refinement of hypotheses.

3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-14 Use lists to simply solutions, generalizing computational problems instead of repeatedly using simple variables.

3A-AP-15	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
3A-AP-18	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
3A-AP-19	Systematically design and develop programs for broad audiences by incorporating feedback from users.
3A-AP-20	Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.
3A-AP-21	Evaluate and refine computational artifacts to make them more usable and accessible.
3A-AP-22	Design and develop computational artifacts working in team roles using collaborative tools.
3A-AP-23	Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
3B-AP-10	Use and adapt classic algorithms to solve computational problems.
3B-AP-11	Evaluate algorithms in terms of their efficiency, correctness, and clarity.
3B-AP-12	Compare and contrast fundamental data structures and their uses.
3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules, and/or objects.
3B-AP-16	Demonstrate code reuse by creating programming solutions using libraries and APIs.
3B-AP-18	Explain security issues that might lead to compromised computer programs.
3B-AP-20	Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.
3B-AP-21	Develop and use a series of test cases to verify that a program performs according to its design specifications.

3B-AP-22	Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
3B-AP-23	Evaluate key qualities of a program through a process such as a code review.
3A-IC-24	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
3A-IC-26	Demonstrate ways a given algorithm applies to problems across disciplines.
3B-IC-25	Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.
3B-IC-28	Debate laws and regulations that impact the development and use of software.

The following Unit Goals align with the 2016 International Society for Technology in Education Standards.

ISTE Empowered Learner (Standard 1)	Students leverage technology to take an active role in choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences.
ISTE Digital Citizen (Standard 2)	Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical.
ISTE Innovative Designer (Standard 4)	Students use a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions.
ISTE Computational Thinker (Standard 5)	Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.
ISTE Creative Communicator (Standard 6)	Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals.

Unit Essential Questions

- How can binary numbers be used to represent all digital data?
- How can algorithms be used to compress data?
- How do variables of both simple and structured data, such as lists, enable us to manage the complexity of a program?

Scope and Sequence

1. Completion of tutorials to build PaintPot and Map Tour apps, and completion of follow-up projects to build creative enhancements to the apps
2. Exploration of methods of image representation, extending the idea that all data can be represented as bits, and exploration of how errors might be detected when data is transmitted
3. Reading and discussion of Chp. 3 of *Blown to Bits*, focusing on binary data as holding more information than what is seen on screen

Assured Assessments

Formative Assessment:
Students will answer interactive self-check questions and add to their ePortfolios answers to reflection questions. They will also create an app, upload it to their ePortfolios, and complete an app write-up. They will also engage in various discussions about technology's impact on transferring information.

Summative Assessment:
Students will take a common end-of-unit assessment scored via a common scoring guide.

Resources

Core

- Abelson, Hal, Ken Ledeen, and Harry Lewis. "Blown to Bits: Your Life, Liberty, and Happiness after the Digital Explosion." http://www.bitsbook.com/. Web.
- MIT App Inventor. ai2.appinventor.mit.edu. Web.
- "Mobile CSP." https://mobilecsp-2017.appspot.com/mobilecsp/course. Web.

Time Allotment

- Approximately 16 school days

UNIT 4
Animation, Simulation, and Modeling

Unit 4 focuses on simulation and modeling; apps are built to simulate randomness, and students investigate the relationship of technology and computing innovations to privacy, the economy, society, and culture.

Unit Goals

At the completion of this unit, students will:

The following Unit Goals align with the 2017 College Board Curriculum Framework for Advanced Placement Computer Science Principles.

1.2.2	Create a computational artifact using computing tools and techniques to solve a problem.
1.2.4	Collaborate in the creation of computational artifacts.
1.3.1	Use computing tools and techniques for creative expression.
2.2.1	Develop an abstraction when writing a program or creating other computational artifacts.
2.2.3	Identify multiple levels of abstractions that are used when writing programs.
2.3.1	Use models and simulations to represent phenomena.
2.3.2	Use models and simulations to formulate, refine, and test hypotheses.
3.3.1	Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information.
4.1.1	Develop an algorithm for implementation in a program.
4.1.2	Express an algorithm in a language.
5.1.1	Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge.
5.1.2	Develop a correct program to solve problems.
5.1.3	Collaborate to develop a program.
5.2.1	Explain how programs implement algorithms.
5.3.1	Use abstraction to manage complexity in programs.
5.5.1	Employ appropriate mathematical and logical concepts in programming.

7.1.1 Explain how computing innovations affect communication, interaction, and cognition.

7.3.1 Analyze the beneficial and harmful effects of computing.

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

The following Unit Goals align with the 2017 Computer Science Teachers' Association (CSTA) Computer Science Standards.

3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

3A-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers.

3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

3B-CS-02 Illustrate ways computing systems implement logic, input, and output through hardware components.

3A-NI-05 Give examples to illustrate how sensitive data can be affected by malware and other attacks.

3A-NI-06 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.

3A-NI-07 Compare various security measures, considering tradeoffs between the usability and security of a computing system.

3A-NI-08 Explain tradeoffs when selecting and implementing cybersecurity recommendations.

3B-NI-04 Compare ways software developers protect devices and information from unauthorized access.

3A-DA-11 Create interactive data visualizations using software tools to help others better understand real-world phenomena.

3A-DA-12 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.

3B-DA-05 Use data analysis tools and techniques to identify patterns in data representing complex systems.

3B-DA-06 Select data collection tools and techniques to generate data sets that support a claim or communicate information.

3B-DA-07	Evaluate the ability of models and simulations to test and support the refinement of hypotheses.
3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
3A-AP-14	Use lists to simply solutions, generalizing computational problems instead of repeatedly using simple variables.
3A-AP-15	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
3A-AP-18	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
3A-AP-19	Systematically design and develop programs for broad audiences by incorporating feedback from users.
3A-AP-23	Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
3B-AP-08	Describe how artificial intelligence drives many software and physical systems.
3B-AP-10	Use and adapt classic algorithms to solve computational problems.
3B-AP-11	Evaluate algorithms in terms of their efficiency, correctness, and clarity.
3B-AP-12	Compare and contrast fundamental data structures and their uses.
3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules, and/or objects.
3B-AP-15	Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.
3B-AP-16	Demonstrate code reuse by creating programming solutions using libraries and APIs.
3B-AP-18	Explain security issues that might lead to compromised computer programs.

3B-AP-20	Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.
3B-AP-21	Develop and use a series of test cases to verify that a program performs according to its design specifications.
3B-AP-22	Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
3B-AP-23	Evaluate key qualities of a program through a process such as a code review.
3A-IC-24	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
3A-IC-25	Test and refine computational artifacts to reduce bias and equity deficits.
3A-IC-26	Demonstrate ways a given algorithm applies to problems across disciplines.
3A-IC-28	Explain the beneficial and harmful effects that intellectual property laws can have on innovation.
3A-IC-29	Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.
3A-IC-30	Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.
3B-IC-25	Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.
3B-IC-27	Predict how computational innovations that have revolutionized aspects of our culture might evolve.
3B-IC-28	Debate laws and regulations that impact the development and use of software.

The following Unit Goals align with the 2016 International Society for Technology in Education Standards.

ISTE Empowered Learner (Standard 1)	Students leverage technology to take an active role in choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences.
ISTE Digital Citizen (Standard 2)	Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical.
ISTE Innovative Designer (Standard 4)	Students use a variety of technologies within a design process to identify and solve problems by creating new,

	useful or imaginative solutions.
ISTE Computational Thinker (Standard 5)	Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.
ISTE Creative Communicator (Standard 6)	Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals.

Unit Essential Questions

- How do computers use simulation and modeling to represent real-world phenomena?
- Why is randomness important, and how is it modeled inside a computer?
- In what ways does simulation and modeling extend our knowledge and benefit society?

Scope and Sequence

1. Completion of tutorials to build Lights Off and Coin Flip apps, and completion of follow-up projects to build creative enhancements to the apps
2. Exploration of the algorithms that dictate how computers can simulate the concept of randomness, and subsequent evaluation of App Inventor's randomness efficiency
3. Reading and discussion of Chp. 2 of *Blown to Bits*, focusing on how the digital explosion affects our privacy, an especially important issue for today's mobile computing generation in which many seem to be willing to trade privacy for convenience

Assured Assessments

Formative Assessment:
Students will answer interactive self-check questions and add to their ePortfolios answers to reflection questions. They will also create an app, upload it to their ePortfolios, and complete an app write-up. They will also engage in various discussions about technology's impact on privacy.

Summative Assessment:
Students will take a common end-of-unit assessment scored via a common scoring guide.

Resources

Core

- Abelson, Hal, Ken Ledeen, and Harry Lewis. "Blown to Bits: Your Life, Liberty, and Happiness after the Digital Explosion." http://www.bitsbook.com/. Web.
- MIT App Inventor. ai2.appinventor.mit.edu. Web.
- "Mobile CSP." https://mobilecsp-2017.appspot.com/mobilecsp/course. Web.

Time Allotment

- Approximately 16 school days

UNIT 5
Create #1: Practice Programming Performance Task

Unit 5 focuses on a practice programming performance task to prepare for a final such task submitted to the College Board.

Unit Goals

At the completion of this unit, students will:

The following Unit Goals align with the 2017 College Board Curriculum Framework for Advanced Placement Computer Science Principles.

1.1.1	Apply a creative development process when creating computational artifacts.
1.2.1	Create a computational artifact for creative expression.
1.2.2	Create a computational artifact using computing tools and techniques to solve a problem.
1.2.4	Collaborate in the creation of computational artifacts.
1.2.5	Analyze the correctness, usability, functionality, and suitability of computational artifacts.
2.2.1	Develop an abstraction when writing a program or creating other computational artifacts.
4.1.1	Develop an algorithm for implementation in a program.
4.1.2	Express an algorithm in a language.
5.1.1	Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge.
5.1.2	Develop a correct program to solve problems.
5.2.1	Explain how programs implement algorithms.
5.3.1	Use abstraction to manage complexity in programs.
5.4.1	Evaluate the correctness of a program.
5.5.1	Employ appropriate mathematical and logical concepts in programming.

The following Unit Goals align with the 2017 Computer Science Teachers' Association (CSTA) Computer Science Standards.

3A-CS-01	Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

3A-CS-02	Compare levels of abstraction and interactions between application software, system software, and hardware layers.
3B-CS-01	Categorize the roles of operating system software.
3B-CS-02	Illustrate ways computing systems implement logic, input, and output through hardware components.
3A-DA-10	Evaluate the tradeoffs in how data elements are organized and where data is stored.
3B-DA-07	Evaluate the ability of models and simulations to test and support the refinement of hypotheses.
3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
3A-AP-14	Use lists to simply solutions, generalizing computational problems instead of repeatedly using simple variables.
3A-AP-15	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
3A-AP-18	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
3A-AP-19	Systematically design and develop programs for broad audiences by incorporating feedback from users.
3A-AP-20	Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.
3A-AP-21	Evaluate and refine computational artifacts to make them more usable and accessible.
3A-AP-22	Design and develop computational artifacts working in team roles using collaborative tools.
3A-AP-23	Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

3B-AP-09	Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.
3B-AP-10	Use and adapt classic algorithms to solve computational problems.
3B-AP-11	Evaluate algorithms in terms of their efficiency, correctness, and clarity.
3B-AP-12	Compare and contrast fundamental data structures and their uses.
3B-AP-13	Illustrate the flow of execution of a recursive algorithm.
3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules, and/or objects.
3B-AP-15	Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.
3B-AP-16	Demonstrate code reuse by creating programming solutions using libraries and APIs.
3B-AP-17	Plan and develop programs for broad audiences using a software life cycle process.
3B-AP-20	Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.
3B-AP-21	Develop and use a series of test cases to verify that a program performs according to its design specifications.
3B-AP-22	Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
3B-AP-23	Evaluate key qualities of a program through a process such as a code review.
3A-IC-24	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
3A-IC-25	Test and refine computational artifacts to reduce bias and equity deficits.
3A-IC-27	Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.
3B-IC-25	Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.

The following Unit Goals align with the 2016 International Society for Technology in Education Standards.

ISTE Empowered Learner	Students leverage technology to take an active role in

(Standard 1)	choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences.
ISTE Digital Citizen (Standard 2)	Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical.
ISTE Knowledge Constructor (Standard 3)	Students critically curate a variety of resources using digital tools to construct knowledge, produce creative artifacts and make meaningful learning experiences for themselves and others.
ISTE Innovative Designer (Standard 4)	Students use a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions.
ISTE Computational Thinker (Standard 5)	Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.
ISTE Creative Communicator (Standard 6)	Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals.
ISTE Global Collaborator (Standard 7)	Students use digital tools to broaden their perspectives and enrich their learning by collaborating with others and working effectively in teams locally and globally.

Unit Essential Questions

- What makes an app socially useful?
- What are the parameters and components of the Create task?
- How do we work collaboratively on a single app?

Scope and Sequence

1. In groups of 2 or 3, students brainstorm an idea for a socially useful app, develop drawing(s) of the User Interface, and create a rough storyboard of how the app will function.
2. Students speak with the teacher to discuss the feasibility of the app; if necessary, they make any changes to their plan.
3. Student groups work collaboratively to develop, test, and debug their app, making sure it meets the following criteria:
    - Documentation of Code: For this task, the app should have well-named components, variables, and procedures. No additional comments are required.
    - Data: For this task, the app should make appropriate use of variables.

- Algorithms: For this task, the app should make appropriate use of sequence and selection control structures.
- Abstraction: For this task, the app should include a programmer-defined procedure.

4. Students keep a journal of problems encountered and how they are resolved.
5. Students create a ePortfolio write-up of their app.
6. Students create a 1-minute video presentation demonstrating their working app.

Assured Assessments

Formative Assessment:
Students will create a working app of their own design, journaling their process, and completing an app write-up in their ePortfolios.

Summative Assessment:
Students' completed apps and video presentations will be scored using the scoring guide of the College Board.

Resources

Core

- College Board. *AP Computer Science Principles Including the Curriculum Framework: Course and Exam Description*. https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf?course=ap-computer-science-principles. Web.
- College Board. *AP Computer Science Principles Create Performance Task Scoring Guidelines*. apcentral.collegeboard.org/pdf/ap-csp-create-performance-task-scoring-guidelines-2019.pdf. Web.
- MIT App Inventor. ai2.appinventor.mit.edu. Web.
- "Mobile CSP." https://mobilecsp-2017.appspot.com/mobilecsp/course. Web.

Time Allotment

- Approximately 10 school days

UNIT 6

Explore #1: Practice Impact of Computing Innovation Performance Task

Unit 6 focuses on a practice impacting of computing innovations performance task to prepare for a final such task submitted to the College Board.

Unit Goals

At the completion of this unit, students will:

The following Unit Goals align with the 2017 College Board Curriculum Framework for Advanced Placement Computer Science Principles.

1.2.1	Create a computational artifact for creative expression.
1.2.2	Create a computational artifact using computing tools and techniques to solve a problem.
3.3.1	Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information.
7.1.1	Explain how computing innovations affect communication, interaction, and cognition.
7.1.2	Explain how people participate in a problem-solving process that scales.
7.2.1	Explain how computing has impacted innovations in other fields.
7.3.1	Analyze the beneficial and harmful effects of computing.
7.4.1	Explain the connections between computing and real-world contexts, including economic, social, and cultural contexts.
7.5.1	Access, manage, and attribute information using effective strategies.
7.5.2	Evaluate online and print sources for appropriateness and credibility.

The following Unit Goals align with the 2017 Computer Science Teachers' Association (CSTA) Computer Science Standards.

3B-CS-01	Categorize the roles of operating system software.
3B-CS-02	Illustrate ways computing systems implement logic, input, and output through hardware components.
3A-NI-05	Give examples to illustrate how sensitive data can be affected by malware and other attacks.
3A-NI-07	Compare various security measures, considering tradeoffs between the usability and security of a computing system.

3B-NI-04	Compare ways software developers protect devices and information from unauthorized access.
3A-DA-10	Evaluate the tradeoffs in how data elements are organized and where data is stored.
3A-DA-11	Create interactive data visualizations using software tools to help others better understand real-world phenomena.
3A-AP-22	Design and develop computational artifacts working in team roles using collaborative tools.
3A-AP-23	Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
3B-AP-08	Describe how artificial intelligence drives many software and physical systems.
3B-AP-18	Explain security issues that might lead to compromised computer programs.
3B-AP-23	Evaluate key qualities of a program through a process such as a code review.
3B-AP-24	Compare multiple programming languages and discuss how their features makes them suitable for solving different types of problems.
3A-IC-24	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
3A-IC-28	Explain the beneficial and harmful effects that intellectual property laws can have on innovation.
3A-IC-29	Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.
3A-IC-30	Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.
3B-IC-25	Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.
3B-IC-26	Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.
3B-IC-27	Predict how computational innovations that have revolutionized aspects of our culture might evolve.
3B-IC-28	Debate laws and regulations that impact the development and use of software.

The following Unit Goals align with the 2016 International Society for Technology in Education Standards.

ISTE Empowered Learner (Standard 1)	Students leverage technology to take an active role in choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences.
ISTE Digital Citizen (Standard 2)	Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical.
ISTE Knowledge Constructor (Standard 3)	Students critically curate a variety of resources using digital tools to construct knowledge, produce creative artifacts and make meaningful learning experiences for themselves and others.
ISTE Creative Communicator (Standard 6)	Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals.
ISTE Global Collaborator (Standard 7)	Students use digital tools to broaden their perspectives and enrich their learning by collaborating with others and working effectively in teams locally and globally.

Unit Essential Questions

- What is a technological innovation?
- How can a single technological innovation impact society in both positive and negative ways?
- How do technological innovations consume, transform, and output data?

Scope and Sequence

1. Individually, students research past technological innovations that have had a major impact on our society, changing the ways we do things.
2. Students speak with the teacher to discuss the innovation they have chosen; if necessary, they make any changes.
3. Students answer the prompts of the College Board regarding the innovation they have chosen:
    - Provide information on the computing innovation and computational artifact.
    - Describe how their computational artifact was created.
    - Explain at least one beneficial effect and at least one harmful effect the computing innovation had.
    - Using specific details, describe the data your innovation uses; how the innovation consumes (as input), produces (as output), and/or transforms data; and at least one data

storage concern, data privacy concern, or data security concern directly related to the computing innovation.
    - Provide a list of at least three online or print resources used to create your computational artifact and/or support your responses through in-text citation.
4. When the document is completed (i.e., all prompts are answered and all sources are cited), each student creates his/her own computational artifact (e.g., music, image, video, infographic, presentation, program, webpage) to express the effects of the chosen innovation.

Assured Assessments

Formative Assessment:
Students will answer all prompts of the College Board regarding the innovation they have chosen.

Summative Assessment:
Students' completed written responses and computational artifacts will be scored using the scoring guide of the College Board.

Resources

Core

- College Board. *AP Computer Science Principles Including the Curriculum Framework: Course and Exam Description*. https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf?course=ap-computer-science-principles. Web.
- College Board. *AP Computer Science Principles Explore Performance Task Scoring Guidelines*. https://apcentral.collegeboard.org/pdf/ap-csp-explore-performance-task-scoring-guidelines-2019.pdf. Web.
- MIT App Inventor. ai2.appinventor.mit.edu. Web.
- "Mobile CSP." https://mobilecsp-2017.appspot.com/mobilecsp/course. Web.

Time Allotment

- Approximately 8 school days

UNIT 7
Algorithms and Procedural Abstraction

Unit 7 focuses on algorithms, procedural abstraction and its advantages, and algorithm efficiency.

Unit Goals

At the completion of this unit, students will:

The following Unit Goals align with the 2017 College Board Curriculum Framework for Advanced Placement Computer Science Principles.

2.2.1	Develop an abstraction when writing a program or creating other computational artifacts.
2.2.2	Use multiple levels of abstraction to write programs.
2.2.3	Identify multiple levels of abstractions that are used when writing programs.
3.2.1	Extract information from data to discover and explain connections or trends.
4.1.1	Develop an algorithm for implementation in a program.
4.1.2	Express an algorithm in a language.
4.2.1	Explain the difference between algorithms that run in a reasonable time and those that do not run in a reasonable time.
4.2.2	Explain the difference between solvable and unsolvable problems in computer science.
4.2.3	Explain the existence of undecidable problems in computer science.
4.2.4	Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity.
5.1.2	Develop a correct program to solve problems.
5.2.1	Explain how programs implement algorithms.
5.3.1	Use abstraction to manage complexity in programs.
5.4.1	Evaluate the correctness of a program.
5.5.1	Employ appropriate mathematical and logical concepts in programming.
7.1.1	Explain how computing innovations affect communication, interaction, and cognition.

7.1.2 Explain how people participate in a problem-solving process that scales.

7.3.1 Analyze the beneficial and harmful effects of computing.

7.5.1 Access, manage, and attribute information using effective strategies.

The following Unit Goals align with the 2017 Computer Science Teachers' Association (CSTA) Computer Science Standards.

3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

3A-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers.

3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

3B-CS-02 Illustrate ways computing systems implement logic, input, and output through hardware components.

3A-DA-11 Create interactive data visualizations using software tools to help others better understand real-world phenomena.

3A-DA-12 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.

3B-DA-05 Use data analysis tools and techniques to identify patterns in data representing complex systems.

3B-DA-06 Select data collection tools and techniques to generate data sets that support a claim or communicate information.

3B-DA-07 Evaluate the ability of models and simulations to test and support the refinement of hypotheses.

3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-14 Use lists to simply solutions, generalizing computational problems instead of repeatedly using simple variables.

3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.

3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.

3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
3A-AP-18	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
3A-AP-19	Systematically design and develop programs for broad audiences by incorporating feedback from users.
3B-AP-09	Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.
3B-AP-10	Use and adapt classic algorithms to solve computational problems.
3B-AP-11	Evaluate algorithms in terms of their efficiency, correctness, and clarity.
3B-AP-12	Compare and contrast fundamental data structures and their uses.
3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules, and/or objects.
3B-AP-15	Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.
3B-AP-16	Demonstrate code reuse by creating programming solutions using libraries and APIs.
3B-AP-17	Plan and develop programs for broad audiences using a software life cycle process.
3B-AP-20	Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.
3B-AP-21	Develop and use a series of test cases to verify that a program performs according to its design specifications.
3B-AP-22	Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
3B-AP-23	Evaluate key qualities of a program through a process such as a code review.
3A-IC-24	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
3A-IC-25	Test and refine computational artifacts to reduce bias and equity deficits.
3A-IC-26	Demonstrate ways a given algorithm applies to problems across disciplines.

The following Unit Goals align with the 2016 International Society for Technology in Education Standards.

ISTE Empowered Learner (Standard 1)	Students leverage technology to take an active role in choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences.
ISTE Digital Citizen (Standard 2)	Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical.
ISTE Knowledge Constructor (Standard 3)	Students critically curate a variety of resources using digital tools to construct knowledge, produce creative artifacts and make meaningful learning experiences for themselves and others.
ISTE Innovative Designer (Standard 4)	Students use a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions.
ISTE Computational Thinker (Standard 5)	Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.
ISTE Creative Communicator (Standard 6)	Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals.

Unit Essential Questions

- How are multiple levels of abstraction used to create computational artifacts?
- In what ways are some algorithms better than others?
- What limits do algorithms have?

Scope and Sequence

1. Collaborative use of apps to classify algorithms experimentally as either logarithmic, linear, n log n, or quadratic
2. Explanation of the concepts of intractability and undecidability through examples of (intractable) problems that cannot be solved efficiently and (unsolvable) problems that cannot be solved at all by means of an algorithm
3. Reading and discussion of Chp. 4 of *Blown to Bits*, focusing on how web searches work

Assured Assessments

Formative Assessment:
Students will answer interactive self-check questions and add to their ePortfolios answers to reflection questions. They will also create an app, upload it to their ePortfolios, and complete an app write-up. They will also engage in various discussions about web searches.

Summative Assessment:
Students will take a common end-of-unit assessment scored via a common scoring guide.

Resources

Core

- Abelson, Hal, Ken Ledeen, and Harry Lewis. "Blown to Bits: Your Life, Liberty, and Happiness after the Digital Explosion." http://www.bitsbook.com/. Web.
- MIT App Inventor. ai2.appinventor.mit.edu. Web.
- "Mobile CSP." https://mobilecsp-2017.appspot.com/mobilecsp/course. Web.

Time Allotment

- Approximately 15 school days

UNIT 8
Communication through the Internet

Unit 8 focuses on the Internet, and its use in apps, as well as an important security concept: cryptography.

Unit Goals

At the completion of this unit, students will:

The following Unit Goals align with the 2017 College Board Curriculum Framework for Advanced Placement Computer Science Principles.

5.1.2 Develop a correct program to solve problems.

5.2.1 Explain how programs implement algorithms.

5.4.1 Evaluate the correctness of a program.

6.1.1 Explain the abstractions in the Internet and how the Internet functions.

6.2.1 Explain characteristics of the Internet and the systems built on it.

6.2.2 Explain how the characteristics of the Internet influence the systems built on it.

6.3.1 Identify existing cybersecurity concerns and potential options to address these issues with the Internet and the systems built on it.

7.3.1 Analyze the beneficial and harmful effects of computing.

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

The following Unit Goals align with the 2017 Computer Science Teachers' Association (CSTA) Computer Science Standards.

3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

3A-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers.

3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

3B-CS-02 Illustrate ways computing systems implement logic, input, and output through hardware components.

3A-NI-04 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.

3A-NI-05	Give examples to illustrate how sensitive data can be affected by malware and other attacks.
3A-NI-06	Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.
3A-NI-07	Compare various security measures, considering tradeoffs between the usability and security of a computing system.
3A-NI-08	Explain tradeoffs when selecting and implementing cybersecurity recommendations.
3B-NI-03	Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).
3B-NI-04	Compare ways software developers protect devices and information from unauthorized access.
3A-DA-09	Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.
3A-DA-11	Create interactive data visualizations using software tools to help others better understand real-world phenomena.
3A-DA-12	Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.
3B-DA-05	Use data analysis tools and techniques to identify patterns in data representing complex systems.
3B-DA-06	Select data collection tools and techniques to generate data sets that support a claim or communicate information.
3B-DA-07	Evaluate the ability of models and simulations to test and support the refinement of hypotheses.
3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
3A-AP-14	Use lists to simply solutions, generalizing computational problems instead of repeatedly using simple variables.
3A-AP-15	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.

3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
3A-AP-18	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
3A-AP-19	Systematically design and develop programs for broad audiences by incorporating feedback from users.
3B-AP-09	Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.
3B-AP-10	Use and adapt classic algorithms to solve computational problems.
3B-AP-11	Evaluate algorithms in terms of their efficiency, correctness, and clarity.
3B-AP-12	Compare and contrast fundamental data structures and their uses.
3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules, and/or objects.
3B-AP-15	Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.
3B-AP-16	Demonstrate code reuse by creating programming solutions using libraries and APIs.
3B-AP-17	Plan and develop programs for broad audiences using a software life cycle process.
3B-AP-20	Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.
3B-AP-21	Develop and use a series of test cases to verify that a program performs according to its design specifications.
3B-AP-22	Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
3B-AP-23	Evaluate key qualities of a program through a process such as a code review.
3A-IC-24	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
3A-IC-25	Test and refine computational artifacts to reduce bias and equity deficits.
3A-IC-26	Demonstrate ways a given algorithm applies to problems across disciplines.

3A-IC-28	Explain the beneficial and harmful effects that intellectual property laws can have on innovation.
3A-IC-29	Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.
3A-IC-30	Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.
3B-IC-26	Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.

The following Unit Goals align with the 2016 International Society for Technology in Education Standards.

ISTE Empowered Learner (Standard 1)	Students leverage technology to take an active role in choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences.
ISTE Digital Citizen (Standard 2)	Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical.
ISTE Knowledge Constructor (Standard 3)	Students critically curate a variety of resources using digital tools to construct knowledge, produce creative artifacts and make meaningful learning experiences for themselves and others.
ISTE Innovative Designer (Standard 4)	Students use a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions.
ISTE Computational Thinker (Standard 5)	Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.
ISTE Creative Communicator (Standard 6)	Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals.
ISTE Global Collaborator (Standard 7)	Students use digital tools to broaden their perspectives and enrich their learning by collaborating with others and working effectively in teams locally and globally.

Unit Essential Questions

- What is the Internet, how is it built, and how does it function?
- What aspects of the Internet's design and development have helped it scale and flourish?

- How is cybersecurity impacting the ever-increasing number of Internet users?

Scope and Sequence

1. Deep exploration of the infrastructure and mechanics of the Internet by examining packet switching, TCP/IP, and the protocol hierarchy
2. Completion of a series of activities using network administration software tools such as Ping and Traceroute
3. After learning about and using some of the basic concepts of cryptography, introduction to the key-exchange problem and then to the basic ideas of public-key encryption as a way of solving the problem
4. Essential details about the Internet's trust system and its implementation in modern browsers to support the exchange of information securely across the internet, focusing on how cybersecurity is made possible through encryption

Assured Assessments

Formative Assessment:
Students will answer interactive self-check questions and add to their ePortfolios answers to reflection questions. They will also create an app, upload it to their ePortfolios, and complete an app write-up. They will also engage in various discussions about cryptography.

Summative Assessment:
Students will take a common end-of-unit assessment scored via a common scoring guide.

Resources

Core

- Abelson, Hal, Ken Ledeen, and Harry Lewis. "Blown to Bits: Your Life, Liberty, and Happiness after the Digital Explosion." http://www.bitsbook.com/. Web.
- MIT App Inventor. ai2.appinventor.mit.edu. Web.
- "Mobile CSP." https://mobilecsp-2017.appspot.com/mobilecsp/course. Web.

Time Allotment

- Approximately 17 school days

UNIT 9
Explore #2: Official Impact of Computing Innovation Performance Task

Unit 9 focuses on the official impact of computing innovations performance task to be submitted to the College Board.

Unit Goals

At the completion of this unit, students will:

The following Unit Goals align with the 2017 College Board Curriculum Framework for Advanced Placement Computer Science Principles.

1.2.1	Create a computational artifact for creative expression.
1.2.2	Create a computational artifact using computing tools and techniques to solve a problem.
3.3.1	Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information.
7.1.1	Explain how computing innovations affect communication, interaction, and cognition.
7.1.2	Explain how people participate in a problem-solving process that scales.
7.2.1	Explain how computing has impacted innovations in other fields.
7.3.1	Analyze the beneficial and harmful effects of computing.
7.4.1	Explain the connections between computing and real-world contexts, including economic, social, and cultural contexts.
7.5.1	Access, manage, and attribute information using effective strategies.
7.5.2	Evaluate online and print sources for appropriateness and credibility.

The following Unit Goals align with the 2017 Computer Science Teachers' Association (CSTA) Computer Science Standards.

3B-CS-01	Categorize the roles of operating system software.
3B-CS-02	Illustrate ways computing systems implement logic, input, and output through hardware components.
3A-NI-05	Give examples to illustrate how sensitive data can be affected by malware and other attacks.
3A-NI-07	Compare various security measures, considering tradeoffs between the usability and security of a computing system.

3B-NI-04	Compare ways software developers protect devices and information from unauthorized access.
3A-DA-10	Evaluate the tradeoffs in how data elements are organized and where data is stored.
3A-DA-11	Create interactive data visualizations using software tools to help others better understand real-world phenomena.
3A-AP-22	Design and develop computational artifacts working in team roles using collaborative tools.
3A-AP-23	Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
3B-AP-08	Describe how artificial intelligence drives many software and physical systems.
3B-AP-18	Explain security issues that might lead to compromised computer programs.
3B-AP-23	Evaluate key qualities of a program through a process such as a code review.
3B-AP-24	Compare multiple programming languages and discuss how their features makes them suitable for solving different types of problems.
3A-IC-24	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
3A-IC-28	Explain the beneficial and harmful effects that intellectual property laws can have on innovation.
3A-IC-29	Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.
3A-IC-30	Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.
3B-IC-25	Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.
3B-IC-26	Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.
3B-IC-27	Predict how computational innovations that have revolutionized aspects of our culture might evolve.
3B-IC-28	Debate laws and regulations that impact the development and use of software.

The following Unit Goals align with the 2016 International Society for Technology in Education Standards.

ISTE Empowered Learner (Standard 1)	Students leverage technology to take an active role in choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences.
ISTE Digital Citizen (Standard 2)	Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical.
ISTE Knowledge Constructor (Standard 3)	Students critically curate a variety of resources using digital tools to construct knowledge, produce creative artifacts and make meaningful learning experiences for themselves and others.
ISTE Creative Communicator (Standard 6)	Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals.
ISTE Global Collaborator (Standard 7)	Students use digital tools to broaden their perspectives and enrich their learning by collaborating with others and working effectively in teams locally and globally.

Unit Essential Questions

- What is a technological innovation?
- How can a single technological innovation impact society in both positive and negative ways?
- How do technological innovations consume, transform, and output data?

Scope and Sequence

1. Individually, students research past technological innovations that have had a major impact on our society, changing the ways we do things.
2. Students speak with the teacher to discuss the innovation they have chosen; if necessary, they make any changes.
3. Students answer the prompts of the College Board regarding the innovation they have chosen:
    - Provide information on the computing innovation and computational artifact.
    - Describe how their computational artifact was created.
    - Explain at least one beneficial effect and at least one harmful effect the computing innovation had.
    - Using specific details, describe the data your innovation uses; how the innovation consumes (as input), produces (as output), and/or transforms data; and at least one data

storage concern, data privacy concern, or data security concern directly related to the computing innovation.
    - Provide a list of at least three online or print resources used to create your computational artifact and/or support your responses through in-text citation.
4. When the document is completed (i.e., all prompts are answered and all sources are cited), each student creates his/her own computational artifact (e.g., music, image, video, infographic, presentation, program, webpage) to express the effects of the chosen innovation.
5. Students upload their written responses and computational artifacts to the College Board Digital Portfolio site.

Assured Assessments

Formative Assessment:
Students will answer all prompts of the College Board regarding the innovation they have chosen.

Summative Assessment:
Students' uploaded written responses and computational artifacts will be scored using the scoring guide of the College Board.

Resources

Core

- College Board. *AP Computer Science Principles Including the Curriculum Framework: Course and Exam Description*. https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf?course=ap-computer-science-principles. Web.
- College Board. *AP Computer Science Principles Explore Performance Task Scoring Guidelines*. https://apcentral.collegeboard.org/pdf/ap-csp-explore-performance-task-scoring-guidelines-2019.pdf. Web.
- MIT App Inventor. ai2.appinventor.mit.edu. Web.
- "Mobile CSP." https://mobilecsp-2017.appspot.com/mobilecsp/course. Web.

Time Allotment

- Approximately 12 school days

UNIT 10
Using and Analyzing Data and Information

Unit 10 focuses on various aspects of using and manipulating data, both within mobile apps and on the Web or Internet; the essential topics of artificial intelligence and machine learning are also discussed.

Unit Goals

At the completion of this unit, students will:

The following Unit Goals align with the 2017 College Board Curriculum Framework for Advanced Placement Computer Science Principles.

1.2.1	Create a computational artifact for creative expression.
1.2.2	Create a computational artifact using computing tools and techniques to solve a problem.
3.1.1	Find patterns and test hypotheses about digitally processed information to gain insight and knowledge.
3.1.2	Collaborate when processing information to gain insight and knowledge.
3.1.3	Explain the insight and knowledge gained from digitally processed data by using appropriate visualizations, notations, and precise language.
3.2.1	Extract information from data to discover and explain connections or trends.
3.2.2	Determine how large data sets impact the use of computational processes to discover information and knowledge.
3.3.1	Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information.
5.3.1	Use abstraction to manage complexity in programs.
5.5.1	Employ appropriate mathematical and logical concepts in programming.
7.1.1	Explain how computing innovations affect communication, interaction, and cognition.
7.2.1	Explain how computing has impacted innovations in other fields.
7.3.1	Analyze the beneficial and harmful effects of computing.

The following Unit Goals align with the 2017 Computer Science Teachers' Association (CSTA) Computer Science Standards.

3A-CS-01	Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.
3A-CS-02	Compare levels of abstraction and interactions between application software, system software, and hardware layers.
3A-CS-03	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
3B-CS-02	Illustrate ways computing systems implement logic, input, and output through hardware components.
3A-NI-05	Give examples to illustrate how sensitive data can be affected by malware and other attacks.
3A-NI-07	Compare various security measures, considering tradeoffs between the usability and security of a computing system.
3A-NI-08	Explain tradeoffs when selecting and implementing cybersecurity recommendations.
3B-NI-03	Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).
3B-NI-04	Compare ways software developers protect devices and information from unauthorized access.
3A-DA-09	Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.
3A-DA-10	Evaluate the tradeoffs in how data elements are organized and where data is stored.
3A-DA-11	Create interactive data visualizations using software tools to help others better understand real-world phenomena.
3A-DA-12	Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.
3B-DA-05	Use data analysis tools and techniques to identify patterns in data representing complex systems.
3B-DA-06	Select data collection tools and techniques to generate data sets that support a claim or communicate information.

3B-DA-07	Evaluate the ability of models and simulations to test and support the refinement of hypotheses.
3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
3A-AP-14	Use lists to simply solutions, generalizing computational problems instead of repeatedly using simple variables.
3A-AP-15	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
3A-AP-18	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
3A-AP-19	Systematically design and develop programs for broad audiences by incorporating feedback from users.
3A-AP-20	Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.
3A-AP-23	Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
3B-AP-08	Describe how artificial intelligence drives many software and physical systems.
3B-AP-09	Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.
3B-AP-10	Use and adapt classic algorithms to solve computational problems.
3B-AP-11	Evaluate algorithms in terms of their efficiency, correctness, and clarity.
3B-AP-12	Compare and contrast fundamental data structures and their uses.
3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules, and/or objects.
3B-AP-15	Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.

3B-AP-16	Demonstrate code reuse by creating programming solutions using libraries and APIs.
3B-AP-17	Plan and develop programs for broad audiences using a software life cycle process.
3B-AP-18	Explain security issues that might lead to compromised computer programs.
3B-AP-20	Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.
3B-AP-21	Develop and use a series of test cases to verify that a program performs according to its design specifications.
3B-AP-22	Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
3B-AP-23	Evaluate key qualities of a program through a process such as a code review.
3A-IC-24	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
3A-IC-28	Explain the beneficial and harmful effects that intellectual property laws can have on innovation.
3A-IC-29	Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.
3A-IC-30	Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.
3B-IC-27	Predict how computational innovations that have revolutionized aspects of our culture might evolve.
3B-IC-28	Debate laws and regulations that impact the development and use of software.

The following Unit Goals align with the 2016 International Society for Technology in Education Standards.

ISTE Empowered Learner (Standard 1)	Students leverage technology to take an active role in choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences.
ISTE Digital Citizen (Standard 2)	Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical.
ISTE Knowledge Constructor	Students critically curate a variety of resources using digital

(Standard 3)	tools to construct knowledge, produce creative artifacts and make meaningful learning experiences for themselves and others.
ISTE Innovative Designer (Standard 4)	Students use a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions.
ISTE Computational Thinker (Standard 5)	Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.
ISTE Global Collaborator (Standard 7)	Students use digital tools to broaden their perspectives and enrich their learning by collaborating with others and working effectively in teams locally and globally.

Unit Essential Questions

- How does continuous access to large amounts of data change how people and organizations make decisions?
- What is the connection between data, information, knowledge, and wisdom?
- How is artificial intelligence and machine learning impacting society and emerging technologies?

Scope and Sequence

1. Building on the idea that data is being recorded about everyone at all times, exploration of the various ways that large data sets can be processed and analyzed, using data visualization tools
2. Creation of an app that transforms a data set into a visualization
3. Creation of an app that stores data in a web database, allowing students to access this data from any app with access to the database
4. Exploration of the fields of artificial intelligence and machine learning through interactive activities

Assured Assessments

Formative Assessment:
Students will answer interactive self-check questions and add to their ePortfolios answers to reflection questions. They will also create apps, upload them to their ePortfolios, and complete app write-ups. They will also engage in various discussions about ownership of digital artifacts and copyright infringement.

Summative Assessment:
Students will take a common end-of-unit assessment scored via a common scoring guide.

Resources

Core

- Abelson, Hal, Ken Ledeen, and Harry Lewis. “Blown to Bits: Your Life, Liberty, and Happiness after the Digital Explosion.” http://www.bitsbook.com/. Web.
- MIT App Inventor. ai2.appinventor.mit.edu. Web.
- “Mobile CSP.” https://mobilecsp-2017.appspot.com/mobilecsp/course. Web.

Time Allotment

- Approximately 14 school days

UNIT 11
Create #2: Official Programming Performance Task

Unit 2 focuses on the official programming performance task to be submitted to the College Board.

Unit Goals

At the completion of this unit, students will:

The following Unit Goals align with the 2017 College Board Curriculum Framework for Advanced Placement Computer Science Principles.

1.1.1 Apply a creative development process when creating computational artifacts.

1.2.1 Create a computational artifact for creative expression.

1.2.2 Create a computational artifact using computing tools and techniques to solve a problem.

1.2.4 Collaborate in the creation of computational artifacts.

1.2.5 Analyze the correctness, usability, functionality, and suitability of computational artifacts.

2.2.1 Develop an abstraction when writing a program or creating other computational artifacts.

4.1.1 Develop an algorithm for implementation in a program.

4.1.2 Express an algorithm in a language.

5.1.1 Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge.

5.1.2 Develop a correct program to solve problems.

5.2.1 Explain how programs implement algorithms.

5.3.1 Use abstraction to manage complexity in programs.

5.4.1 Evaluate the correctness of a program.

5.5.1 Employ appropriate mathematical and logical concepts in programming.

The following Unit Goals align with the 2017 Computer Science Teachers' Association (CSTA) Computer Science Standards.

3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

3A-CS-02	Compare levels of abstraction and interactions between application software, system software, and hardware layers.
3B-CS-01	Categorize the roles of operating system software.
3B-CS-02	Illustrate ways computing systems implement logic, input, and output through hardware components.
3A-DA-10	Evaluate the tradeoffs in how data elements are organized and where data is stored.
3B-DA-07	Evaluate the ability of models and simulations to test and support the refinement of hypotheses.
3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
3A-AP-14	Use lists to simply solutions, generalizing computational problems instead of repeatedly using simple variables.
3A-AP-15	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
3A-AP-18	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
3A-AP-19	Systematically design and develop programs for broad audiences by incorporating feedback from users.
3A-AP-20	Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.
3A-AP-21	Evaluate and refine computational artifacts to make them more usable and accessible.
3A-AP-22	Design and develop computational artifacts working in team roles using collaborative tools.
3A-AP-23	Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

3B-AP-09	Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.
3B-AP-10	Use and adapt classic algorithms to solve computational problems.
3B-AP-11	Evaluate algorithms in terms of their efficiency, correctness, and clarity.
3B-AP-12	Compare and contrast fundamental data structures and their uses.
3B-AP-13	Illustrate the flow of execution of a recursive algorithm.
3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules, and/or objects.
3B-AP-15	Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.
3B-AP-16	Demonstrate code reuse by creating programming solutions using libraries and APIs.
3B-AP-17	Plan and develop programs for broad audiences using a software life cycle process.
3B-AP-20	Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.
3B-AP-21	Develop and use a series of test cases to verify that a program performs according to its design specifications.
3B-AP-22	Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
3B-AP-23	Evaluate key qualities of a program through a process such as a code review.
3A-IC-24	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
3A-IC-25	Test and refine computational artifacts to reduce bias and equity deficits.
3A-IC-27	Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.
3B-IC-25	Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.

The following Unit Goals align with the 2016 International Society for Technology in Education Standards.

ISTE Empowered Learner	Students leverage technology to take an active role in

(Standard 1)	choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences.
ISTE Digital Citizen (Standard 2)	Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical.
ISTE Knowledge Constructor (Standard 3)	Students critically curate a variety of resources using digital tools to construct knowledge, produce creative artifacts and make meaningful learning experiences for themselves and others.
ISTE Innovative Designer (Standard 4)	Students use a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions.
ISTE Computational Thinker (Standard 5)	Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.
ISTE Creative Communicator (Standard 6)	Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals.
ISTE Global Collaborator (Standard 7)	Students use digital tools to broaden their perspectives and enrich their learning by collaborating with others and working effectively in teams locally and globally.

Unit Essential Questions

- What makes an app socially useful?
- What are the parameters and components of the Create task?
- How do we work collaboratively on a single app?

Scope and Sequence

1. In groups of 2 or 3, students brainstorm an idea for a socially useful app, develop drawing(s) of the User Interface, and create a rough storyboard of how the app will function.
2. Students speak with the teacher to discuss the feasibility of the app; if necessary, they make any changes to their plan.
3. Student groups work collaboratively to develop, test, and debug their app, making sure it meets the following criteria:
    - Documentation of Code: For this task, the app should have well-named components, variables, and procedures. No additional comments are required.
    - Data: For this task, the app should make appropriate use of variables.

- Algorithms: For this task, the app should make appropriate use of sequence and selection control structures.
- Abstraction: For this task, the app should include a programmer-defined procedure.

4. Students keep a journal of problems encountered and how they are resolved.
5. Students create a ePortfolio write-up of their app.
6. Students create a 1-minute video presentation demonstrating their working app.
7. Students upload their written responses and video presentations to the College Board Digital Portfolio site.

Assured Assessments

Formative Assessment:
Students will create a working app of their own design, journaling their process, and completing an app write-up in their ePortfolios.

Summative Assessment:
Students' uploaded written responses and video presentations will be scored using the scoring guide of the College Board.

Resources

Core

- College Board. *AP Computer Science Principles Including the Curriculum Framework: Course and Exam Description*. https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf?course=ap-computer-science-principles. Web.
- College Board. *AP Computer Science Principles Create Performance Task Scoring Guidelines*. apcentral.collegeboard.org/pdf/ap-csp-create-performance-task-scoring-guidelines-2019.pdf. Web.
- MIT App Inventor. ai2.appinventor.mit.edu. Web.
- "Mobile CSP." https://mobilecsp-2017.appspot.com/mobilecsp/course. Web.

Time Allotment

- Approximately 17 school days

UNIT 12
College Board Examination Preparation

Unit 12 focuses on the details of the College Board Examination in Advanced Placement Computer Science Principles; its goals are all the goals of the prior units.

Unit Essential Questions

- What is the format, style, and length of the College Board Examination in Advanced Placement Computer Science Principles?
- What types of programming languages will be included on the Examination?
- What is the AP CSP Exam Reference Sheet, and how can I successfully use it?

Scope and Sequence

1. The AP Computer Science Principles Examination assesses both the application of computational thinking practices and the understanding of related big ideas. Questions may be accompanied by non-textual stimulus material such as diagrams, charts, or other graphical illustrations.
2. No designated programming language is included in the exam; instead, a pseudocode is used. The AP CSP Exam Reference Sheet is provided during the exam to help students understand that pseudocode format and the meaning of the exam questions. Students will review key vocabulary and participate in tracing, a technique used to simulate a dry run through the code or pseudocode line by line by hand as if one is the computer executing the code; tracing can be used for debugging, proving that a program runs correctly, or figuring out what a code actually does.
3. Students will complete at least one full released Examination during class; if time permits, they will complete another.

Assured Assessments
Formative Assessment:
Students will participate in practice questions and released items related to the College Board Examination.

Summative Assessment:
Students will participate in a full released College Board Examination.

Resources

Core

- College Board. "AP Computer Science Principles Exam Reference Sheet." http://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf#page=125. Web.
- "Mobile CSP." https://mobilecsp-2017.appspot.com/mobilecsp/course. Web.

Time Allotment

- Approximately 9 school days

UNIT 13
Beyond the Exam

Unit 13 includes the opportunity for students to experience some additional app tutorials.

Unit Goals

At the completion of this unit, students will:

The following Unit Goals align with the 2017 College Board Curriculum Framework for Advanced Placement Computer Science Principles.

1.1.1	Apply a creative development process when creating computational artifacts.
1.2.1	Create a computational artifact for creative expression.
1.2.2	Create a computational artifact using computing tools and techniques to solve a problem.
1.2.4	Collaborate in the creation of computational artifacts.
1.2.5	Analyze the correctness, usability, functionality, and suitability of computational artifacts.
2.2.1	Develop an abstraction when writing a program or creating other computational artifacts.
4.1.1	Develop an algorithm for implementation in a program.
4.1.2	Express an algorithm in a language.
5.1.1	Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge.
5.1.2	Develop a correct program to solve problems.
5.2.1	Explain how programs implement algorithms.
5.3.1	Use abstraction to manage complexity in programs.
5.4.1	Evaluate the correctness of a program.
5.5.1	Employ appropriate mathematical and logical concepts in programming.

The following Unit Goals align with the 2017 Computer Science Teachers' Association (CSTA) Computer Science Standards.

3A-CS-01	Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

3A-CS-02	Compare levels of abstraction and interactions between application software, system software, and hardware layers.
3B-CS-01	Categorize the roles of operating system software.
3B-CS-02	Illustrate ways computing systems implement logic, input, and output through hardware components.
3A-DA-10	Evaluate the tradeoffs in how data elements are organized and where data is stored.
3B-DA-07	Evaluate the ability of models and simulations to test and support the refinement of hypotheses.
3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
3A-AP-14	Use lists to simply solutions, generalizing computational problems instead of repeatedly using simple variables.
3A-AP-15	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
3A-AP-18	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
3A-AP-19	Systematically design and develop programs for broad audiences by incorporating feedback from users.
3A-AP-20	Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.
3A-AP-21	Evaluate and refine computational artifacts to make them more usable and accessible.
3A-AP-22	Design and develop computational artifacts working in team roles using collaborative tools.
3A-AP-23	Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

3B-AP-09	Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.
3B-AP-10	Use and adapt classic algorithms to solve computational problems.
3B-AP-11	Evaluate algorithms in terms of their efficiency, correctness, and clarity.
3B-AP-12	Compare and contrast fundamental data structures and their uses.
3B-AP-13	Illustrate the flow of execution of a recursive algorithm.
3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules, and/or objects.
3B-AP-15	Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.
3B-AP-16	Demonstrate code reuse by creating programming solutions using libraries and APIs.
3B-AP-17	Plan and develop programs for broad audiences using a software life cycle process.
3B-AP-20	Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.
3B-AP-21	Develop and use a series of test cases to verify that a program performs according to its design specifications.
3B-AP-22	Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
3B-AP-23	Evaluate key qualities of a program through a process such as a code review.
3A-IC-24	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
3A-IC-25	Test and refine computational artifacts to reduce bias and equity deficits.
3A-IC-27	Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.
3B-IC-25	Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.

The following Unit Goals align with the 2016 International Society for Technology in Education Standards.

ISTE Empowered Learner	Students leverage technology to take an active role in

(Standard 1)	choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences.
ISTE Digital Citizen (Standard 2)	Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical.
ISTE Knowledge Constructor (Standard 3)	Students critically curate a variety of resources using digital tools to construct knowledge, produce creative artifacts and make meaningful learning experiences for themselves and others.
ISTE Innovative Designer (Standard 4)	Students use a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions.
ISTE Computational Thinker (Standard 5)	Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.
ISTE Creative Communicator (Standard 6)	Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals.
ISTE Global Collaborator (Standard 7)	Students use digital tools to broaden their perspectives and enrich their learning by collaborating with others and working effectively in teams locally and globally.

Unit Essential Questions

- What are some other functions of App Inventor?
- What are possible career paths that utilize programming?

Scope and Sequence

1. Creation of assigned apps, building and enhancing them
2. Discussion of where to go next, including how to continue exploring computer science

Assured Assessments

Formative Assessment:
For each app created, students will upload the app to their ePortfolios and complete an app write-up.

Summative Assessment:
There is no summative assessment for this final unit.

Resources

Core

- MIT App Inventor. ai2.appinventor.mit.edu. Web.
- "Mobile CSP." https://mobilecsp-2017.appspot.com/mobilecsp/course. Web.

Time Allotment

- Approximately 12 school days

COURSE CREDIT

One credit in STEM
One class period daily for a full year

PREREQUISITES

Successful completion of ACP Geometry.

SUPPLEMENTARY MATERIALS/RESOURCES/TECHNOLOGY

Department- and teacher-prepared materials

Chromebook or laptop with the Chrome web browser

Android device (phone or tablet) or a laptop with an Android emulator

College Board website including past Advanced Placement Computer Science Principles tests

CURRENT REFERENCES

College Board: Advanced Placement Computer Science Principles:

https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf?course=ap-computer-science-principles

Mobile CSP:

https://mobilecsp-2017.appspot.com/mobilecsp/course

ASSURED STUDENT PERFORMANCE RUBRICS

- Trumbull High School School-Wide Writing Rubric (attached)
- Trumbull High School School-Wide Problem-Solving Rubric (attached)
- Trumbull High School School-Wide Independent Learning and Thinking Rubric (attached)

Trumbull High School School-Wide Writing Rubric

Category/ Weight	Exemplary 4 Student work:	Goal 3 Student work:	Working Toward Goal 2 Student work:	Needs Support 1-0 Student work:
Purpose X_____	• Establishes and maintains a clear purpose • Demonstrates an insightful understanding of audience and task	• Establishes and maintains a purpose • Demonstrates an accurate awareness of audience and task	• Establishes a purpose • Demonstrates an awareness of audience and task	• Does not establish a clear purpose • Demonstrates limited/no awareness of audience and task
Organization X_____	• Reflects sophisticated organization throughout • Demonstrates logical progression of ideas • Maintains a clear focus • Utilizes effective transitions	• Reflects organization throughout • Demonstrates logical progression of ideas • Maintains a focus • Utilizes transitions	• Reflects some organization throughout • Demonstrates logical progression of ideas at times • Maintains a vague focus • May utilize some ineffective transitions	• Reflects little/no organization • Lacks logical progression of ideas • Maintains little/no focus • Utilizes ineffective or no transitions
Content X_____	• Is accurate, explicit, and vivid • Exhibits ideas that are highly developed and enhanced by specific details and examples	• Is accurate and relevant • Exhibits ideas that are developed and supported by details and examples	• May contain some inaccuracies • Exhibits ideas that are partially supported by details and examples	• Is inaccurate and unclear • Exhibits limited/no ideas supported by specific details and examples
Use of Language X_____	• Demonstrates excellent use of language • Demonstrates a highly effective use of standard writing that enhances communication • Contains few or no errors. Errors do not detract from meaning	• Demonstrates competent use of language • Demonstrates effective use of standard writing conventions • Contains few errors Most errors do not detract from meaning	• Demonstrates use of language • Demonstrates use of standard writing conventions • Contains errors that detract from meaning	• Demonstrates limited competency in use of language • Demonstrates limited use of standard writing conventions • Contains errors that make it difficult to determine meaning

Trumbull High School School-Wide Problem-Solving Rubric

Category/ Weight	Exemplary 4	Goal 3	Working Toward Goal 2	Needs Support 1-0
Understanding X_____	• Student demonstrates clear understanding of the problem and the complexities of the task	• Student demonstrates sufficient understanding of the problem and most of the complexities of the task	• Student demonstrates some understanding of the problem but requires assistance to complete the task	• Student demonstrates limited or no understanding of the fundamental problem after assistance with the task
Research X_____	• Student gathers compelling information from multiple sources including digital, print, and interpersonal	• Student gathers sufficient information from multiple sources including digital, print, and interpersonal	• Student gathers some information from few sources including digital, print, and interpersonal	• Student gathers limited or no information
Reasoning and Strategies X_____	• Student demonstrates strong critical thinking skills to develop a comprehensive plan integrating multiple strategies	• Student demonstrates sufficient critical thinking skills to develop a cohesive plan integrating strategies	• Student demonstrates some critical thinking skills to develop a plan integrating some strategies	• Student demonstrates limited or no critical thinking skills and no plan
Final Product and/or Presentation X_____	• Solution shows deep understanding of the problem and its components • Solution shows extensive use of 21st-century technology skills	• Solution shows sufficient understanding of the problem and its components • Solution shows sufficient use of 21st-century technology skills	• Solution shows some understanding of the problem and its components • Solution shows some use of 21st-century technology skills	• Solution shows limited or no understanding of the problem and its components • Solution shows limited or no use of 21st-century technology skills

Trumbull High School School-Wide
Independent Learning and Thinking Rubric

Category/ Weight	Exemplary 4	Goal 3	Working Toward Goal 2	Needs Support 1-0
Proposal X_____	• Student demonstrates a strong sense of initiative by generating compelling questions, creating uniquely original projects/work	• Student demonstrates initiative by generating appropriate questions, creating original projects/work	• Student demonstrates some initiative by generating questions, creating appropriate projects/work	• Student demonstrates limited or no initiative by generating few questions and creating projects/work
Independent Research & Development X_____	• Student is analytical, insightful, and works independently to reach a solution	• Student is analytical, and works productively to reach a solution	• Student reaches a solution with direction	• Student is unable to reach a solution without consistent assistance
Presentation of Final Product X_____	• Presentation shows compelling evidence of an independent learner and thinker • Solution shows deep understanding of the problem and its components • Solution shows extensive and appropriate application of 21st-century skills	• Presentation shows clear evidence of an independent learner and thinker • Solution shows adequate understanding of the problem and its components • Solution shows adequate application of 21st-century skills	• Presentation shows some evidence of an independent learner and thinker • Solution shows some understanding of the problem and its components • Solution shows some application of 21st-century skills	• Presentation shows limited or no evidence of an independent learner and thinker • Solution shows limited or no understanding of the problem and its components • Solution shows limited or no application of 21st-century skills