AP® Computer Science Principles

Course Planning and Pacing Guide

Andrew Kuemmel
West High School ▸ Madison, WI

About the College Board

The College Board is a mission-driven not-for-profit organization that connects students to college success and opportunity. Founded in 1900, the College Board was created to expand access to higher education. Today, the membership association is made up of over 6,000 of the world's leading educational institutions and is dedicated to promoting excellence and equity in education. Each year, the College Board helps more than seven million students prepare for a successful transition to college through programs and services in college readiness and college success — including the SAT® and the Advanced Placement Program®. The organization also serves the education community through research and advocacy on behalf of students, educators, and schools. For further information, visit **www.collegeboard.org**.

AP® Equity and Access Policy

The College Board strongly encourages educators to make equitable access a guiding principle for their AP® programs by giving all willing and academically prepared students the opportunity to participate in AP. We encourage the elimination of barriers that restrict access to AP for students from ethnic, racial, and socioeconomic groups that have been traditionally underrepresented. Schools should make every effort to ensure their AP classes reflect the diversity of their student population. The College Board also believes that all students should have access to academically challenging course work before they enroll in AP classes, which can prepare them for AP success. It is only through a commitment to equitable preparation and access that true equity and excellence can be achieved.

Welcome to the AP Computer Science Principles Course Planning and Pacing Guides

This guide is one of several course planning and pacing guides designed for AP® Computer Science Principles teachers. Each provides an example of how to design instruction for the AP course based on the author's teaching context (e.g., demographics, schedule, school type, setting). These course planning and pacing guides highlight how the components of the *AP Computer Science Principles Curriculum Framework* — including the learning objectives, essential knowledge statements, and computational thinking practices — are addressed in the course. Each guide also provides valuable suggestions for teaching the course, including the selection of resources, instructional activities, and classroom assessments. The authors have offered insight into the *why* and *how* behind their instructional choices — displayed along the right side of the individual unit plans — to aid in course planning for AP Computer Science Principles teachers.

The primary purpose of these comprehensive guides is to model approaches for planning and pacing curriculum throughout the school year. However, they can also help with syllabus development when used in conjunction with the resources created to support the AP Course Audit: the Syllabus Development Guide and the four Annotated Sample Syllabi. These resources include samples of evidence and illustrate a variety of strategies for meeting curricular requirements.

Contents

Instructional Setting

West High School ▸ Madison, WI

School

West High School is a comprehensive urban high school located near the University of Wisconsin – Madison, with an enrollment of approximately 2,100 students. More than 50 different languages are spoken by the students of West High School, and our families come from a wide range of income levels. In 2013 the school administered 435 AP® Exams in 23 different areas.

Student population

The student population is approximately:

- 53 percent white
- 17 percent Hispanic
- 12 percent African American
- 11 percent Asian
- 6 percent multiracial
- 1 percent Native American or other

Approximately 22 percent of our students are English language learners, 35 percent are low-income students, and 15 percent are special education students.

Instructional time

The Madison Metropolitan School District begins the school year on or after September 1. Our academic calendar is 182 days, including testing days. We have seven periods in a day, and class periods last 45 minutes on Mondays and 52 minutes on other days. There are effectively 140 teaching days from the start of the school year until the administration of AP Exams in May.

Student preparation

Our school has three full-year computer science courses. Our Introduction to Programming course enrolls approximately 50–60 students and teaches programming in Scratch, App Inventor, and Lego NXT, as well as problem solving and computer hardware. AP Computer Science Principles and AP Computer Science A each enroll approximately 40–45 students. Even though we offer a three-year sequence, few students have room in their schedules to take all three courses.

In AP Computer Science Principles, about one third of the students have completed the Introduction to Programming course, and another third were exposed to some kind of extracurricular computer experience in their middle school years. The final third of the students have had no formal computer science experience.

Over half of the students in the course are in 10th grade, with approximately a quarter in 9th grade and the rest in 11th or 12th grade.

Instructional Setting (continued)

Primary planning resources

I have no formal planning resource, although the content of the course was inspired by lectures and assignments from the University of Wisconsin – Madison Introduction to Computation course, designed by Professor Andrea Arpaci-Dusseau (**http://pages.cs.wisc.edu/~dusseau/Classes/CS202-F11**).

Large portions of the programming curriculum are taken from the Beauty and Joy of Computing course created by the University of California, Berkeley (**http://bjc.berkeley.edu/website/curriculum.html**).

Overview of the Course

AP Computer Science Principles provides students from a wide range of backgrounds the opportunity to understand and participate in the dramatic changes to our lives brought about by computing. I have four underlying goals for students in this course: First, I want my students to experience an inspiring survey of computer science, to learn to think like a computer scientist, and to consider how computational thinking can enhance their eventual fields of study. Second, I want to demystify the magic behind many of the devices and applications that my students use every day. Seeing how computational artifacts were designed helps students consider the biases and limitations of computing and provides inspiration for the assignments that students do. Third, I want to create a base of information and skills that my students will use in their adult lives in areas such as intelligent systems, data collection and privacy, and the infrastructure of the Internet. Finally, I want my students to understand that we have choices in how we use computing. Students enter the course with ideas of why they like to use technology, how technology should be used, and what role they will (or will not) have in the creation of technology. After the course I want them to know that they can be the creators of computing, and that the use of computing tools changes us as individuals and as a society.

Reading and writing skills are key elements of this course and must be taught, not assumed. My students read to gain knowledge of how to create and use software, as well as how to analyze the impact of computing on society. Writing helps them clarify their thinking and is an integral part of the AP Computer Science Principles performance tasks. To build their reading and writing skills, my students follow a sequence of watching a video, having a discussion, reading about the topic, taking notes or marking up the article, discussing the article as a class, and then writing a personal opinion about the topic.

When organizing the course around the big ideas in the curriculum framework, I focus each week on one creative innovation. We read about the innovation and discuss its impact and use of the Internet. Students think creatively to design similar innovations that do not yet exist. We discuss the data this innovation takes as input and produces as output. We consider the algorithms and abstractions of the innovation. We do some "unplugged" or "plugged" activities to demonstrate algorithms. Finally, students create a program that relates to the innovation. This approach requires careful sequencing to pair innovations with programming concepts.

Students in this course learn by constructing their own knowledge. Solving problems their way rather than memorizing solutions leads to stronger learning, more motivation, and more success by underrepresented groups. For instance, my students solve intractable problems, design ways to deliver information on a distributed network, and create their own sorting algorithms.

We start programming in Scratch. This programming environment encourages creativity, reduces frustration, and helps students believe that they can make anything they can dream of. Most of the programming assignments are differentiated and evaluated as formative assessments. Students who do not complete assignments during class time may come into the lab during lunch or before school.

In my class, students are not expected to complete computer-based assignments at home. This policy helps all students succeed in the course regardless of the computing resources at home.

Computational Thinking Practices

P1: Connecting Computing

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. Students in this course study these effects, and they learn to draw connections between different computing concepts. Students are expected to:

- Identify impacts of computing;
- Describe connections between people and computing; and
- Explain connections between computing concepts.

P2: Creating Computational Artifacts

Computing is a creative discipline in which creation takes many forms, such as remixing digital music, generating animations, developing websites, and writing programs. Students in this course 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. Students are expected to:

- Create an artifact with a practical, personal, or societal intent;
- Select appropriate techniques to develop a computational artifact; and
- Use appropriate algorithmic and information-management principles.

P3: Abstracting

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. Students in this course 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. Students are expected to:

- Explain how data, information, or knowledge is represented for computational use;
- Explain how abstractions are used in computation or modeling;
- Identify abstractions; and
- Describe modeling in a computational context.

P4: Analyzing Problems and Artifacts

The results and artifacts of computation and the computational techniques and strategies that generate them can be understood both intrinsically 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. Students in this course design and produce solutions, models, and artifacts, and they evaluate and analyze their own computational work as well as the computational work others have produced. Students are expected to:

- Evaluate a proposed solution to a problem;
- Locate and correct errors;
- Explain how an artifact functions; and
- Justify appropriateness and correctness.

P5: Communicating

Students in this course 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. Communication includes written and oral descriptions supported by graphs, visualizations, and computational analysis. Students are expected to:

- Explain the meaning of a result in context;
- Describe computation with accurate and precise language, notations, or visualizations; and
- Summarize the purpose of a computational artifact.

P6: Collaborating

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. Students in this course collaborate on a number of activities, including investigation of questions using data sets and in the production of computational artifacts. Students are expected to:

- 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 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; and
- Review and revise their work as needed to create a high-quality artifact.

Pacing Overview

Unit	Hours of Instruction	Unit Summary
1: Introduction to Computer Science Principles	12	To begin, students are introduced to the course. To set the tone for the course, students reflect on the computing innovation that has had the most impact on their lives and then consider that innovation in light of the seven big ideas. Students write simple algorithms, learn about different kinds of programming languages, and write simple programs in Scratch with sound, art, and graphics. Finally, students consider the imperative "program or be programmed" in light of social media websites.
2: The Internet	17	In this unit, students increase in their understanding of the Internet through the other six big ideas. Students learn the algorithms of search engines, the abstractions of the Internet, and the connection between cybersecurity and the design of the Internet. The unit ends with two projects that build foundational skills for the performance tasks and a test.
3: Artificial Intelligence	17	In the third unit, students expand upon all seven big ideas through a discussion of artificial intelligence. Starting with the Turing test, students consider creative new uses of intelligence and consider how computers can be programmed to process information in useful ways. Students learn about the impact of algorithms in all aspects of society, and they analyze the abstractions and algorithms in some famous computers that have defeated humans in competitions such as trivia and chess.
4: Abstraction and Simulation	17	In this unit, students learn about how binary numbers form the foundation for all digital information. They see how logic gates are combined to add two numbers and to store data, and they discuss the concepts behind operating systems and central processing units. During this unit we switch over to Snap! and start making custom blocks with inputs that return values.
5: Data	20	The second semester begins with a unit on how data powers computing innovations. Students learn to program with lists and practice different sorting and searching algorithms by hand. They consider different aspects of big data and investigate publicly available data sets. In this unit, students use the Python programming language to manipulate lists.
6: Intractable Problems	8	This unit connects ideas of algorithms with ideas of security. Along the way it discusses heuristics, cryptography, why certain problems are not solvable in a reasonable amount of time, and why some problems are not solvable at all. In this unit, students use the Python programming language to manipulate strings.

Pacing Overview (continued)

Unit	Hours of Instruction	Unit Summary
7. Impact on Society	8	The beneficial and harmful effects of computing have been a part of every unit in this course up to this point, but in this unit, students develop some formal ways of evaluating computing's impact on society. Education, income, political power, class, work, and family are just some of the societal areas that have been affected by computing innovations. As students grow and mature, I want them to see beyond their own experiences and interests to appreciate the impacts of computing on their peers, communities, and countries, as well as on our global society.

UNIT 1: INTRODUCTION TO COMPUTER SCIENCE PRINCIPLES

Estimated Time: 12 Hours

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 4.1, EU 5.1, EU 7.1, EU 7.2, EU 7.3, EU 7.5

Projects and Major Assignments:

- Personal Reflection on Impact
- Algorithmic Puzzles
- Program or Be Programmed
- Music Machine
- Computational Art

Guiding Questions

- **How is programming a form of expression?**
- **What does the phrase "program or be programmed" mean to you?**
- **How does programming impact society?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 7.1.1: Explain how computing innovations affect communication, interaction, and cognition. [P4]		**Formative Assessment: Impact on Your Life** On the first day of class, I ask students, *What computing innovation has had the most impact on your life?* Students consider the question individually, in small groups, and as a class. That night they have a conversation with an adult in their life and ask the adult the same question. The next day, students report back with the answers, creating a document that includes both their response and the adult's response. On the third day, I present the seven big ideas to students. They complete a chart that provides an example of each big idea as it relates to the innovation they chose and a chart that does the same thing for the innovation their adult chose. **Essential knowledge addressed: 7.1.1** L-O
LO 4.1.2: Express an algorithm in a language. [P5] **LO 7.5.1:** Access, manage, and attribute information using effective strategies. [P1]	**Web** "River Crossing Puzzle"	**Instructional Activity: What Is an Algorithm?** Students perform a Web search to find and record two different definitions of *algorithm*, one of which must be from a university website. They practice writing algorithms for tasks such as "How to find a partner in this class" or "How to find a parking spot" at a crowded place in our city. The lesson concludes with students solving various online algorithmic puzzles, such as filling vases with water and crossing a river with pets. **Essential knowledge addressed: 4.1.2** B, G; **7.5.1**B

Usually there are organizational tasks that take place in the first few days of school, and this activity easily fits into any schedule.

I start the year with this assessment because it provides me knowledge about each student's interests, background knowledge, family, and skill level while giving the students a feel for the scope of the course. The feedback I give to students may encourage them to write more, use complete sentences, or add detail.

UNIT 1: INTRODUCTION TO COMPUTER SCIENCE PRINCIPLES

Estimated Time: 12 Hours

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 4.1, EU 5.1, EU 7.1, EU 7.2, EU 7.3, EU 7.5

Projects and Major Assignments:

- Personal Reflection on Impact
- Algorithmic Puzzles
- Program or Be Programmed
- Music Machine
- Computational Art

Guiding Questions

- **How is programming a form of expression?**
- **What does the phrase "program or be programmed" mean to you?**
- **How does programming impact society?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 4.1.2: Express an algorithm in a language. [P5] **LO 7.5.1:** Access, manage, and attribute information using effective strategies. [P1]	**Web** "Frogs" "Jacquard Loom Walkthrough" "Lightbot 2.0" "Player Piano 'The Entertainer'"	**Instructional Activity: What Is a Program?** Students perform a Web search to find and record two different definitions of *programming language*, one of which must be from a university website. In a short lecture, I give an overview of different levels of programming languages (e.g., physical, low level, high level, visual, pseudocode). I then show pictures and/or videos of physical languages, such as "Jacquard Loom Walkthrough" and the player piano video. Next, students practice playing an online puzzle in which frogs jump over each other; they then write down a simple programming language to solve that puzzle. If there is time, students play Lightbot. **Essential knowledge addressed: 4.1.2** A, C-E; **7.5.1**B
LO 7.3.1: Analyze the beneficial and harmful effects of computing. [P4]	**Web** "Program or Be Programmed"	**Instructional Activity: Program or Be Programmed** Students fill out a note sheet in which they write about their favorite social media site and explain what they think the purpose of the site is. Then we watch a video clip promoting the book *Program or Be Programmed*. Afterward, we read a public excerpt from the book and have a class discussion about the topics raised. At the end of class, students write about how they agree or disagree with Rushkoff's statement "you either make the software or you become the software." **Essential knowledge addressed: 7.3.1** A, J

Exposing students to bold ideas is important in an Advanced Placement course. Whether students agree with an idea or not, they will gain skills in analyzing arguments and resolving them with their own opinions.

UNIT 1: INTRODUCTION TO COMPUTER SCIENCE PRINCIPLES

Estimated Time: 12 Hours

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 4.1, EU 5.1, EU 7.1, EU 7.2, EU 7.3, EU 7.5

Projects and Major Assignments:

- Personal Reflection on Impact
- Algorithmic Puzzles
- Program or Be Programmed
- Music Machine
- Computational Art

Guiding Questions

- **How is programming a form of expression?**
- **What does the phrase "program or be programmed" mean to you?**
- **How does programming impact society?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 7.3.1: Analyze the beneficial and harmful effects of computing. [P4] **LO 7.5.1:** Access, manage, and attribute information using effective strategies. [P1] **LO 7.5.2:** Evaluate online and print sources for appropriateness and credibility. [P5]	**Web** Manjoo, "A Bright Side to Facebook's Experiments on Its Users"	**Instructional Activity: Experiments on Social Media Users** To follow up on the idea of the previous lesson that "you either make the software or you become the software," we read aloud an article about how Facebook and other social media sites perform experiments on their users, and students answer questions about the article. Then students search for an additional article on the Internet about this topic, cite their source, and provide a brief summary. **Essential knowledge addressed: 7.3.1** A, J; **7.5.1** A-C; **7.5.2** A, B	*At the start of the previous lesson, students were asked about their favorite social media site. After they have read Ruskhoff and the article about social media experiments on users, they may see these sites in a new light. This process of personal growth will continue throughout the year as we explore the impact of computing.*
LO 1.2.1: Create a computational artifact for creative expression. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.3.1:** Use computing tools and techniques for creative expression. [P2] **LO 7.2.1:** Explain how computing has impacted innovations in other fields. [P1]	**Web** McKinley, "Central Park, the Soundtrack" "Pärt Inspired Musical Live Coding Etude Performed in Scratch" "Scratch Etude - Pärt - SIGCSE Initial Performance Code"	**Instructional Activity: Programming Is a Form of Expression** In this Web-augmented lecture, students consider different ways that people express themselves through programming. Students see examples of mobile apps, such as location-based albums apps, music-making apps, and live-coding apps. Next, students write a journal entry, responding to the prompt, *If you could make a location-based playlist, what would it have on it?* To conclude the activity, students practice playing a Scratch project that demonstrates live coding, and they modify the project to make their own music. **Essential knowledge addressed: 1.2.1** A-B, **1.2.3** A-C, **1.3.1**B, **7.2.1**G	*Students show creativity not only in the creation of artifacts and in the solving of problems but also in the description or design of things that they would like to make.* *I use journal entries to encourage students to think creatively, give their opinions, or pose questions they have about computer science. My students seem to write more words when they use word processing for their entries (as opposed to handwriting).*

UNIT 1: INTRODUCTION TO COMPUTER SCIENCE PRINCIPLES

Estimated Time: 12 Hours

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 4.1, EU 5.1, EU 7.1, EU 7.2, EU 7.3, EU 7.5

Projects and Major Assignments:

- Personal Reflection on Impact
- Algorithmic Puzzles
- Program or Be Programmed
- Music Machine
- Computational Art

Guiding Questions

- How is programming a form of expression?
- What does the phrase "program or be programmed" mean to you?
- How does programming impact society?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.2.1: Create a computational artifact for creative expression. [P2] **LO 5.1.1:** Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. [P2]		**Instructional Activity: Making Music** Using Scratch, students learn how to make sprites and how to make those sprites make interactive music. In the process, they use repeat loops and event-driven control structures. **Essential knowledge addressed: 1.2.1** D, E; **5.1.1**B
LO 1.2.1: Create a computational artifact for creative expression. [P2] **LO 5.1.1:** Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. [P2]	**Web** "Computation Arts"	**Instructional Activity: Making Art** Over two days, students watch a short video that demonstrates computational art. They learn how to use the pen commands and the random block in Scratch, and they learn how to make art by having a sprite follow keyboard and mouse commands. **Essential knowledge addressed: 1.2.1** A-C, **5.1.1**B

These programming assignments are structured so that students with different levels of programming experience can be successful. In addition, the student work produced informs me about each of my student's interests, level of programming knowledge, and ability to problem solve.

UNIT 1: INTRODUCTION TO COMPUTER SCIENCE PRINCIPLES

Estimated Time: 12 Hours

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 4.1, EU 5.1, EU 7.1, EU 7.2, EU 7.3, EU 7.5

Projects and Major Assignments:

- Personal Reflection on Impact
- Algorithmic Puzzles
- Program or Be Programmed
- Music Machine
- Computational Art

Guiding Questions

- How is programming a form of expression?
- What does the phrase "program or be programmed" mean to you?
- How does programming impact society?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.2.1: Create a computational artifact for creative expression. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.3.1:** Use computing tools and techniques for creative expression. [P2] **LO 7.1.1:** Explain how computing innovations affect communication, interaction, and cognition. [P4] **LO 7.2.1:** Explain how computing has impacted innovations in other fields. [P1] **LO 7.3.1:** Analyze the beneficial and harmful effects of computing. [P4]		**Summative Assessment: Short Essay** Students write a short essay about the phrase "programming is a form of expression" or about the topic "program or be programmed." I remind them to use examples of things we have discussed in this unit and encourage them to include a connection to computing's impact on society. Students have one class period to work on their paper, but they can work on it outside of class if they wish. **Essential knowledge addressed: 1.2.1** A, B; **1.2.3** A-C; **1.3.1**B; **7.1.1** L-0; **7.2.1**G; **7.3.1** A, J

This summative assessment addresses all of the guiding questions for this unit.

UNIT 2: THE INTERNET

Estimated Time: 17 Hours

BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 6 The Internet
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.2, EU 3.1, EU 3.2, EU 4.1, EU 5.2, EU 5.4, EU 6.1, EU 6.2, EU 6.3, EU 7.1, EU 7.2, EU 7.3, EU 7.4, EU 7.5

Projects and Major Assignments:

- Privacy on the Web
- Simulating the PageRank Algorithm
- Internet Explanation in Scratch
- Is the Internet Broken?

Guiding Questions

- **How do search engines work?**
- **How do agreements help the Internet work?**
- **How are today's cybersecurity concerns consequences of decisions made when the Internet was designed?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 3.1.1: Find patterns and test hypotheses about digitally processed information to gain insight and knowledge. [P4] **LO 4.1.2:** Express an algorithm in a language. [P5]		**Instructional Activity: Web Crawlers** To start this unit, students work in small groups to discuss the question, *How are search engines so good at what they do?* Small-group notes are reported out to the large group. Students then do a Web quest to find a definition of *Web crawler*. Next, I use a series of Google Docs to create my own Web of pages and links. The pages do not have any content besides links to other pages. Students are challenged to use a starting link to "crawl" my Web of pages. Their final product is a visual map of the pages and their connections. **Essential knowledge addressed: 3.1.1**C, **4.1.2**B
LO 3.2.1: Extract information from data to discover and explain connections or trends. [P1] **LO 7.1.1:** Explain how computing innovations affect communication, interaction, and cognition. [P4] **LO 7.2.1:** Explain how computing has impacted innovations in other fields. [P1]	**Print** MacCormick, chapter 2	**Instructional Activity: Indexing** To follow on the previous day's activity, students practice making an index of key words on a series of Web pages that I have created. Students learn how a Web index not only records the words on a page but also their location on the page. The metadata created allows for searches for words that are next to each other. Finally, students are provided with an index, which is an abstraction of the actual content of the page. They must answer queries based only on the index, without seeing the original page. **Essential knowledge addressed: 3.2.1** H, I; **7.1.1** M, N; **7.2.1**C
LO 3.2.1: Extract information from data to discover and explain connections or trends. [P1] **LO 7.1.1:** Explain how computing innovations affect communication, interaction, and cognition. [P4] **LO 7.2.1:** Explain how computing has impacted innovations in other fields. [P1]	**Print** MacCormick, chapter 3	**Instructional Activity: Ranking Web Pages** As a warm-up, students perform a specific online query with at least four words. They record the top two hits from three different search engines. Then, using a series of sample pages of text, students perform Web rankings using the following techniques: ▶ Frequency of the word ▶ Number of incoming links to a page ▶ Reputation of each page ▶ Web surfer algorithm **Essential knowledge addressed: 3.2.1** H, I; **7.1.1** M, N; **7.2.1**C

UNIT 2: THE INTERNET

Estimated Time: 17 Hours

BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 6 The Internet
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.2, EU 3.1, EU 3.2, EU 4.1, EU 5.2, EU 5.4, EU 6.1, EU 6.2, EU 6.3, EU 7.1, EU 7.2, EU 7.3, EU 7.4, EU 7.5

Projects and Major Assignments:

- Privacy on the Web
- Simulating the PageRank Algorithm
- Internet Explanation in Scratch
- Is the Internet Broken?

Guiding Questions

- **How do search engines work?**
- **How do agreements help the Internet work?**
- **How are today's cybersecurity concerns consequences of decisions made when the Internet was designed?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 7.1.1: Explain how computing innovations affect communication, interaction, and cognition. [P4] **LO 7.3.1:** Analyze the beneficial and harmful effects of computing. [P4]		**Formative Assessment: Privacy on the Web** Working in pairs, students choose one of the following topics to research during class: proxy servers, cookies, right to be forgotten, search engines as trend predictors, or some similar issue regarding the impact of search engines. After 15 minutes of research, each pair gives a 1-minute report to the class. **Essential knowledge addressed: 7.1.1** G, M, 0; **7.3.1** G-M
LO 6.1.1: Explain the abstractions in the Internet and how the Internet functions. [P3] **LO 6.2.1:** Explain characteristics of the Internet and the systems built on it. [P5] **LO 6.2.2:** Explain how the characteristics of the Internet influence the systems built on it. [P4] **LO 7.4.1:** Explain the connections between computing and real-world contexts, including economic, social, and cultural contexts. [P1]	**Web** "How Does the Internet Work?"	**Instructional Activity: Internet Origins, Spirit, and Governance** As an opening, unplugged activity, students turn to their partners and answer the question, *Who is in charge of the Internet?* Then students research and write down the meaning and role of the following organizations: Internet Corporation for Assigned Names and Numbers (ICANN), Internet Engineering Task Force (IETF), and Internet Society (ISOC). Next, the class watches a video explaining how the Internet works, and they can define the following: *Internet protocol*, *Internet*, *IP address*, *and Internet exchange.* Students think of a single word that explains why the Internet continues to work so well; they share their answers in small groups or we discuss them as a class (possible answer: agreements). Students end the class by researching definitions of the following terms: *protocols*, *open standards*, and *interfaces.* **Essential knowledge addressed: 6.1.1** A-F, I; **6.2.1** A, D; **6.2.2** D, E, H; **7.4.1**E

As students work, I give feedback on their searching techniques. I also provide feedback via comments given after the presentations.

In a large class, if there isn't time for the presentations, pairs could write a short summary into a shared Google Doc instead.

BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 6 The Internet
BIG IDEA 7 Global Impact

Essential Understandings:

▶ EU 1.2, EU 3.1, EU 3.2, EU 4.1, EU 5.2, EU 5.4, EU 6.1, EU 6.2, EU 6.3, EU 7.1, EU 7.2, EU 7.3, EU 7.4, EU 7.5

Projects and Major Assignments:

▶ Privacy on the Web ▶ Simulating the PageRank Algorithm ▶ Internet Explanation in Scratch ▶ Is the Internet Broken?

Guiding Questions

▶ How do search engines work? ▶ How do agreements help the Internet work? ▶ How are today's cybersecurity concerns consequences of decisions made when the Internet was designed?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 6.2.1: Explain characteristics of the Internet and the systems built on it. [P5] **LO 6.2.2:** Explain how the characteristics of the Internet influence the systems built on it. [P4]	**Web** "There and Back Again: A Packet's Tale. How Does the Internet Work?"	**Instructional Activity: Simulating TCP/IP** Students watch a video explaining how packets travel around the TCP/IP. They look up definitions of *IP address*, *packet*, *router*, and *TCI/IP*. Next, as a class, students simulate the transmission of packets through routers on the Internet. Students at the edge have an IP address, and students in the middle of the class act as the routers. As I walk through the room, I act to slow down routers or destroy packets. **Essential knowledge addressed: 6.2.1** A, C, D; **6.2.2** B, F, G	*For this and the next two activities, more detailed step-by-step instructions are provided in the* AP Computer Science Principles Internet Curriculum Module.
LO 6.1.1: Explain the abstractions in the Internet and how the Internet functions. [P3] **LO 6.2.1:** Explain characteristics of the Internet and the systems built on it. [P5] **LO 6.2.2:** Explain how the characteristics of the Internet influence the systems built on it. [P4]	**Web** "How Domain Name Servers Work"	**Instructional Activity: Domain Name Servers** Students go through an activity that demonstrates how domain name servers work. One student acts as the domain name server, and the other students act as individual Internet users. The individual users write a domain name (e.g., collegeboard.org) on a piece of paper and pass it to the person who is the server. That person turns the domain name into an IP address (e.g., 128.23.01.22). The class pauses and discusses how this might actually happen on the Internet and how a single system could scale to handle the large number of domain names and users on the Internet. Finally, students visit a website that shows how domain name servers work. **Essential knowledge addressed: 6.1.1**G; **6.2.1**B; **6.2.2** A, C	
LO 6.2.1: Explain characteristics of the Internet and the systems built on it. [P5] **LO 6.2.2:** Explain how the characteristics of the Internet influence the systems built on it. [P4]		**Instructional Activity: Routing Tables** In this activity, students consider how routers know where to send packets on the Internet. Students are given a piece of paper showing a distributed network of approximately 20 computers linked together, labeled A, B, C, D, etc. Students use the map to fill in a table that shows, for any given computer on the map, where a given packet should be passed along in order to reach its final destination. **Essential knowledge addressed: 6.2.1** A, C, D; **6.2.2** B, F, G	*Routing tables are not directly part of the curriculum framework, but students frequently ask me how the routers "know" where to send the packets.*

UNIT 2: THE INTERNET

Estimated Time: 17 Hours

BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 6 The Internet
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.2, EU 3.1, EU 3.2, EU 4.1, EU 5.2, EU 5.4, EU 6.1, EU 6.2, EU 6.3, EU 7.1, EU 7.2, EU 7.3, EU 7.4, EU 7.5

Projects and Major Assignments:

- Privacy on the Web
- Simulating the PageRank Algorithm
- Internet Explanation in Scratch
- Is the Internet Broken?

Guiding Questions

- How do search engines work?
- How do agreements help the Internet work?
- How are today's cybersecurity concerns consequences of decisions made when the Internet was designed?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 5.2.1: Explain how programs implement algorithms. [P3] **LO 5.4.1:** Evaluate the correctness of a program. [P4]		**Instructional Activity: Coding** Students create a brief Scratch project to practice selection statements with `IF`. In the project, the user inputs a domain name and the program outputs an IP address. Broadcast and receive blocks are used to pass the message along to other parts of the program, thus simulating the distributed nature of the domain name system. **Essential knowledge addressed: 5.2.1** A, B, D; **5.4.1** D, G	
LO 6.1.1: Explain the abstractions in the Internet and how the Internet functions. [P3] **LO 6.2.2:** Explain how the characteristics of the Internet influence the systems built on it. [P4]		**Instructional Activity: Scaling and Net Neutrality** Students begin class by writing a journal entry on the question, *What are some problems that you think may arise as the Internet expands?* The class holds a brief discussion about IPv6 and other topics that come up. We then discuss the issue of net neutrality. I ask students to find two different articles and one video about the subject. We then watch one video and hold a class discussion during which students stand on one side of the room or the other to show their position on net neutrality. **Essential knowledge addressed: 6.1.1**H, **6.2.2** I-K	*This lesson builds students' understanding of the Internet by relating to topics of their choice.* *If no students bring up the issue of running out of IP addresses under the IPv4 protocol, I make sure to discuss this problem and its solution, IPv6.*
LO 6.3.1: Identify existing cybersecurity concerns and potential options to address these issues with the Internet and the systems built on it. [P1]	**Web** "Norse – IPViking Live"	**Instructional Activity: Cybersecurity in the News** In this lesson, I present a recent event regarding cybersecurity and we discuss it as a class. Students start by sharing what they have heard about the cybersecurity event. Then I give them an article to read. Students mark up the article, putting boxes around terms that we have learned in this unit and circling terminology that they do not understand. At the end of class, students visit a website that visualizes cyberattacks in real time. **Essential knowledge addressed: 6.3.1** A, C, D, F, G	*By using recent events, I find that my students have interesting contributions to add to the class discussion that I may not have provided myself. In 2012, we discussed the Stuxnet Virus. In 2013, we discussed Edward Snowden and the National Security Agency. In 2014, we discussed the Sony and Target data hacks.*

UNIT 2: THE INTERNET

Estimated Time: 17 Hours

BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 6 The Internet
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.2, EU 3.1, EU 3.2, EU 4.1, EU 5.2, EU 5.4, EU 6.1, EU 6.2, EU 6.3, EU 7.1, EU 7.2, EU 7.3, EU 7.4, EU 7.5

Projects and Major Assignments:

- Privacy on the Web
- Simulating the PageRank Algorithm
- Internet Explanation in Scratch
- Is the Internet Broken?

Guiding Questions

▸ How do search engines work? ▸ How do agreements help the Internet work? ▸ How are today's cybersecurity concerns consequences of decisions made when the Internet was designed?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 6.1.1: Explain the abstractions in the Internet and how the Internet functions. [P3] **LO 6.3.1:** Identify existing cybersecurity concerns and potential options to address these issues with the Internet and the systems built on it. [P1] **LO 7.5.1:** Access, manage, and attribute information using effective strategies. [P1] **LO 7.5.2:** Evaluate online and print sources for appropriateness and credibility. [P5]		**Formative Assessment: Is the Internet Broken?** After discussing recent cyberattacks, students independently research articles that relate the lack of security on the Internet to the very design features that enable its widespread use. Students write one or two paragraphs discussing the connection between these activities and the design of the Internet, citing their sources. **Essential knowledge addressed: 6.1.1**D; **6.3.1** A, B; **7.5.1** A-C; **7.5.2** A, B
LO 6.3.1: Identify existing cybersecurity concerns and potential options to address these issues with the Internet and the systems built on it. [P1]	**Web** "How the Cyberattack on Spamhaus Unfolded" Markoff and Perlroth, "Attacks Used Internet Against Itself to Clog Traffic"	**Instructional Activity: Acting Out a DDoS Attack** Students read about a recent DDoS attack that affected large parts of the Internet. While individually reading the printed-out article, students highlight terms that we have discussed in this unit, as well as terms that they do not understand. Next, the class acts out a DDoS attack. I act as the server under attack, one person acts as a legitimate user, and one person acts as the attacker. The rest of the students act as unsuspecting home computers that have been taken over by the attacker. The attacker writes a question on the board that he or she wants all the affected computers to ask me (the server). Then all the students (except the attacker) stand up and approach me to ask their question at the same time. When the legitimate user tries to ask me his or her question, I am too overwhelmed to respond. Finally, to summarize the lesson, students discuss what a metaphor for a DDoS attack might be if each student in the school acted as an infected computer. **Essential knowledge addressed: 6.3.1** B, E

This formative assessment provides me knowledge of student learning in this unit, as well as knowledge of students' ability to research and write. I return the paragraphs to students the next day and provide information about their writing and citation.

UNIT 2: THE INTERNET

Estimated Time: 17 Hours

BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 6 The Internet
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.2, EU 3.1, EU 3.2, EU 4.1, EU 5.2, EU 5.4, EU 6.1, EU 6.2, EU 6.3, EU 7.1, EU 7.2, EU 7.3, EU 7.4, EU 7.5

Projects and Major Assignments:

- Privacy on the Web
- Simulating the PageRank Algorithm
- Internet Explanation in Scratch
- Is the Internet Broken?

Guiding Questions

- **How do search engines work?**
- **How do agreements help the Internet work?**
- **How are today's cybersecurity concerns consequences of decisions made when the Internet was designed?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.2.2: Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6]		**Formative Assessment: Internet Explanation in Scratch** Over one or two days, working with a partner, students make a Scratch program that demonstrates something about the Internet. I encourage them to use their story-making skills in Scratch, including broadcast and receive, costume changes, and background changes. **Essential knowledge addressed: 1.2.2**B; **1.2.4** A, C-F
LO 7.3.1: Analyze the beneficial and harmful effects of computing. [P4]	**Web** "United States of Secrets (Part Two)"	**Instructional Activity: Privacy and the Government** Students conduct a Web search to find resources for exploring how governments — in particular, agencies such as the National Security Agency — are finding ways to collect information about their citizens. Students scan the results of the search and make tabs in their browser of articles they find informative. After about 5–10 minutes, students give a 15-second oral report on some interesting finding to the class. Students watch "United States of Secrets (Part Two)" and write a journal entry that responds to the question, *What information should governments be allowed to collect about their citizens?* **Essential knowledge addressed: 7.3.1** G, H, J, L

Writing a computer program to convey knowledge is a nontraditional technique. This helps students see how programming can be a creative activity. I assess their work in class by watching students run the program, and I provide feedback by commenting on their programming techniques, content, and use of Scratch's artistic tools.

UNIT 2: THE INTERNET

Estimated Time: 17 Hours

BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 6 The Internet
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.2, EU 3.1, EU 3.2, EU 4.1, EU 5.2, EU 5.4, EU 6.1, EU 6.2, EU 6.3, EU 7.1, EU 7.2, EU 7.3, EU 7.4, EU 7.5

Projects and Major Assignments:

- Privacy on the Web
- Simulating the PageRank Algorithm
- Internet Explanation in Scratch
- Is the Internet Broken?

Guiding Questions

- **How do search engines work?**
- **How do agreements help the Internet work?**
- **How are today's cybersecurity concerns consequences of decisions made when the Internet was designed?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives in this unit are addressed.		**Summative Assessment: Internet Paper and Unit Quiz** Working with a partner, students choose a topic that relates to the ideas of this unit, they conduct research, and they write a short paper. Their paper must have at least two relevant Web sources. In addition, students complete a written quiz of short-answer questions that focus on the key terms and ideas of this unit. **All of the unit's essential knowledge statements are addressed.**

This summative assessment addresses all of the guiding questions for this unit.

UNIT 3: ARTIFICIAL INTELLIGENCE

Estimated Time: 17 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

▶ EU 1.2, EU 2.2, EU 4.1, EU 4.2, EU 5.1, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3

Projects and Major Assignments:

▶ Decision Tree Program ▶ Game of Stones ▶ Rock-Paper-Scissors ▶ Innovations in Robotics Paper ▶ Betting Algorithm for *Jeopardy!*

Guiding Questions

▶ How do computers act in intelligent ways? ▶ How do we define artificial intelligence? ▶ How can algorithms be written to win games? ▶ How have competitions between humans and computers defined what intelligence means?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.2.5: Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4]		**Instructional Activity: First Impressions** Students participate in a think-pair-share activity to answer the question, *What is the first thing that comes to mind when you hear the phrase "artificial intelligence"?* Next, I present a definition of *artificial intelligence* in light of the Turing test. Students individually research different Internet chatbots, interact with them, and write about the features of the best and worst conversations. **Essential knowledge addressed: 1.2.5** A-D
LO 7.1.2: Explain how people participate in a problem-solving process that scales. [P4] **LO 7.2.1:** Explain how computing has impacted innovations in other fields. [P1]		**Instructional Activity: Modern Turing Tests: Newspaper Articles** More and more news stories are being written by algorithms, and in this activity, students individually research this topic and share an example with the class. They brainstorm other possible "modern-day Turing tests" in small groups. Next, students research reverse Turing tests and CAPTCHAs and record their findings in the class learning management system (Moodle). **Essential knowledge addressed: 7.1.2** C, E; **7.2.1**C
LO 7.1.2: Explain how people participate in a problem-solving process that scales. [P4] **LO 7.2.1:** Explain how computing has impacted innovations in other fields. [P1]	**Print** MacCormack, chapter 9 **Web** "Course Introduction - Stanford NLP - Professor Dan Jurafsky & Chris Manning" NACLO practice problems	**Instructional Activity: AI Then and Now: Old Goals, New Branches, Machine Learning, and Neuro-Linguistic Programming (NLP)** Students compare and contrast the old-school goal of AI (to build a machine that understands the world) with the new-school goal of AI (to do useful things). Students do a machine-learning activity by hand; they are given a number of pictures and a set of rules, and they classify the pictures into one of four categories. The next day, students watch a video about natural language processing, think up phrases that are difficult for a computer to parse, and try sample contest problems in computational linguistics. **Essential knowledge addressed: 7.1.2**A; **7.2.1** C, G

To encourage writing, I try to find topics that students are eager to write about. I also precede writing with momentum-building activities.

One interesting extension to this activity might be to give students four news stories, two of which were written by an algorithm. Can students identify which stories were written by humans?

UNIT 3: ARTIFICIAL INTELLIGENCE

Estimated Time: 17 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

▸ EU 1.2, EU 2.2, EU 4.1, EU 4.2, EU 5.1, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3

Projects and Major Assignments:

▸ Decision Tree Program ▸ Game of Stones ▸ Rock-Paper-Scissors ▸ Innovations in Robotics Paper ▸ Betting Algorithm for *Jeopardy!*

Guiding Questions

▸ How do computers act in intelligent ways? ▸ How do we define artificial intelligence? ▸ How can algorithms be written to win games? ▸ How have competitions between humans and computers defined what intelligence means?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.2.5: Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **LO 7.3.1:** Analyze the beneficial and harmful effects of computing. [P4]	**Web** "Algorithms are Taking Over the World: Christopher Steiner at TEDx OrangeCoast"	**Instructional Activity: Are Algorithms Taking Over?** Students watch a TED talk that describes different ways that algorithms are being used behind the scenes. The class has a debate about whether society's movement toward algorithmic decision making is good or bad. **Essential knowledge addressed: 1.2.5** A, B; **7.3.1** J, K
LO 4.1.1: Develop an algorithm for implementation in a program. [P2] **LO 5.1.2:** Develop a correct program to solve problems. [P2] **LO 5.4.1:** Evaluate the correctness of a program. [P4]		**Formative Assessment: Programming Decision Trees** Many useful websites make recommendations to users based on a series of questions. In this activity, each student creates a decision tree that will recommend, for instance, a movie, based on what kinds of genres and characters the user likes. Students use sprites to ask the questions and sprite costumes to show the final recommendations. **Essential knowledge addressed: 4.1.1**C; **5.1.2** A-C; **5.4.1** C-E, G

Students are motivated in this assessment by choosing an area in which they have expertise, such as movies, music, or games. The assessment can be differentiated by the number of branches and by how user input is handled. I assess their work in class by watching students run the program, and I provide feedback by commenting on their programming techniques, their content, and the depth of the tree.

UNIT 3: ARTIFICIAL INTELLIGENCE

Estimated Time: 17 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.2, EU 2.2, EU 4.1, EU 4.2, EU 5.1, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3

Projects and Major Assignments:

- Decision Tree Program
- Game of Stones
- Rock-Paper-Scissors
- Innovations in Robotics Paper
- Betting Algorithm for *Jeopardy!*

Guiding Questions

- How do computers act in intelligent ways?
- How do we define artificial intelligence?
- How can algorithms be written to win games?
- How have competitions between humans and computers defined what intelligence means?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 5.5.1: Employ appropriate mathematical and logical concepts in programming. [P1]		**Instructional Activity: Modulo Arithmetic** Students complete a worksheet that builds their understanding of modulo arithmetic. In the worksheet, students work through a sequence of adding time on a clock ("What time is 5 hours after 10 a.m.?"), performing integer division ("What is left over when 27 is divided by 3?"), and then using the mod operator. Students practice in class with about a dozen examples and then do another dozen problems for homework. **Essential knowledge addressed: 5.5.1** A, D
LO 5.1.2: Develop a correct program to solve problems. [P2]		**Instructional Activity: Game of Stones** One way that computers win games against humans is by knowing a mathematical algorithm that leads to victory. In this activity, students play a game in which the computer and the user take turns removing one, two, or three stones from a pile. The player who takes the last stone wins. **Essential knowledge addressed: 5.1.2** A-C
LO 2.2.1: Develop an abstraction when writing a program or creating other computational artifacts. [P2] **LO 4.1.1:** Develop an algorithm for implementation in a program. [P2] **LO 5.1.2:** Develop a correct program to solve problems. [P2]	**Web** "Rock-Paper-Scissors: You vs. the Computer"	**Instructional Activity: Rock-Paper-Scissors** Students play the rock-paper-scissors game against an online computer and discuss the computer's choosing algorithm. Then I teach them a more advanced version of the game, called Rock Paper Scissors Lizard Spock, and they play it in class against each other. Next, as a whole class, students write an algorithm that represents the five words as integers and order the choices in such a way that a single line of code can determine the winner. This helps them learn how to think like a computer scientist. The next day, students implement their algorithm as a Scratch program. **Essential knowledge addressed: 2.2.1** A-C; **4.1.1** H, I; **5.1.2** D-F, I

This activity can be differentiated to be made more challenging by requiring the player to enter only one, two, or three stones and also by creating a random initial pile that always favors the computer.

The RPSLS program has a deeper complexity than previous programs, and students may need guidance. I like to give students a list of milestones for programs like this to help me track their progress and help them see success along the way.

UNIT 3: ARTIFICIAL INTELLIGENCE

Estimated Time: 17 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

▶ EU 1.2, EU 2.2, EU 4.1, EU 4.2, EU 5.1, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3

Projects and Major Assignments:

▶ Decision Tree Program ▶ Game of Stones ▶ Rock-Paper-Scissors ▶ Innovations in Robotics Paper ▶ Betting Algorithm for *Jeopardy!*

Guiding Questions

▶ **How do computers act in intelligent ways?** ▶ **How do we define artificial intelligence?** ▶ **How can algorithms be written to win games?** ▶ **How have competitions between humans and computers defined what intelligence means?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 4.2.4: Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity. [P4]	**Web** "Chess Exhibit" "Exploring the Epic Chess Match of Our Time"	**Instructional Activity: IBM's Deep Blue and the Minimax Algorithm** Students read an interactive explanation of the history of AI and chess and in the process learn the minimax algorithm. They then read an article and watch a video about Garry Kasparov's matches with Deep Blue and the aftermath. **Essential knowledge addressed: 4.2.4** A, D
LO 7.2.1: Explain how computing has impacted innovations in other fields. [P1]	**Web** "The Watson Trivia Challenge" "What Is Watson?"	**Instructional Activity: Watson** Students play an online game similar to *Jeopardy!* against an algorithm called Watson. Also at that resource, they watch a video about Watson's uses and read an article about its creation. Finally, students research Watson's current uses by visiting the IBM Watson website. **Essential knowledge addressed: 7.2.1** B, C
LO 5.4.1: Evaluate the correctness of a program. [P4]		**Formative Assessment: Programming Watson's Betting Algorithm** Students discuss the rules of the Final Jeopardy round of *Jeopardy!*, in which each player makes a wager between $0 and their current dollar amount. After playing a few sample rounds of Final Jeopardy as a class, each student writes down his or her own Final Jeopardy betting algorithm in words. They code this algorithm, run different situations, and compare the answers that their algorithms give. **Essential knowledge addressed: 5.4.1** F, G

The nature of creating their own algorithm allows for excellent differentiation. I challenge my advanced students to develop a sophisticated algorithm.

Students get feedback on their algorithms from me as I test them out personally and from their peers during the activity.

UNIT 3: ARTIFICIAL INTELLIGENCE

Estimated Time: 17 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.2, EU 2.2, EU 4.1, EU 4.2, EU 5.1, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3

Projects and Major Assignments:

▸ Decision Tree Program ▸ Game of Stones ▸ Rock-Paper-Scissors ▸ Innovations in Robotics Paper ▸ Betting Algorithm for *Jeopardy!*

Guiding Questions

▸ **How do computers act in intelligent ways?** ▸ **How do we define artificial intelligence?** ▸ **How can algorithms be written to win games?** ▸ **How have competitions between humans and computers defined what intelligence means?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.2.5: Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4]	**Web** "CS247: Human-Computer Interaction Design Studio"	**Instructional Activity: Human–Computer Interaction** Students do a quick Internet search for "human–computer interaction" (HCI), share their results out loud, receive an overview of HCI, and watch a video describing it. In pairs, they identify one computational artifact from their experience that they believe has good design and one that has bad design. Next, they research two different articles about current trends in human–robot interaction. **Essential knowledge addressed: 1.2.5** A-D
LO 7.1.1: Explain how computing innovations affect communication, interaction, and cognition. [P4] **LO 7.2.1:** Explain how computing has impacted innovations in other fields. [P1]	**Web** Hoffman, "Reading Pain in a Human Face"	**Instructional Activity: Social Intelligence** Students read the *New York Times* article and visit the interactive quiz linked on that page ("Are These People in Real Pain or Just Faking It?"), which shows how computers are gaining the ability to determine emotion from video. Students then write a paragraph in the school learning system (Moodle) about what intelligence a toddler has about other humans that a computer does not — this is another way to define social intelligence. **Essential knowledge addressed: 7.1.1**L, **7.2.1**B
LO 5.1.2: Develop a correct program to solve problems. [P2] **LO 5.4.1:** Evaluate the correctness of a program. [P4]		**Instructional Activity: What Day Is Today?** Students write a program that reports the date in common language, such as "Friday, January 8th." In the process students discuss why it is important to test an algorithm for correctness. They write iterative loops to test every possible date in one year. **Essential knowledge addressed: 5.1.2**J, **5.4.1** I-K

The research part of this activity is a prelude to the Explore Performance Task. I want to gather information about how students research a topic and what sources they choose. I provide feedback to individuals and to the class as a whole about their research techniques.

UNIT 3: ARTIFICIAL INTELLIGENCE

Estimated Time: 17 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

▶ EU 1.2, EU 2.2, EU 4.1, EU 4.2, EU 5.1, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3

Projects and Major Assignments:

▶ Decision Tree Program ▶ Game of Stones ▶ Rock-Paper-Scissors ▶ Innovations in Robotics Paper ▶ Betting Algorithm for *Jeopardy!*

Guiding Questions

▶ **How do computers act in intelligent ways?** ▶ **How do we define artificial intelligence?** ▶ **How can algorithms be written to win games?** ▶ **How have competitions between humans and computers defined what intelligence means?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives in this unit are addressed.		**Summative Assessment: Unit Quiz** Students take a quiz consisting of approximately 10 short-answer questions, such as *How can a computer tell if a person is smiling?* or *What are two methods that computers use to win games against humans?* **All of the unit's essential knowledge statements are addressed.**

This summative assessment addresses all of the guiding questions for this unit.

UNIT 4: ABSTRACTION AND SIMULATION

Estimated Time: 21 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 5 Programming
BIG IDEA 6 The Internet

Essential Understandings:

- EU 1.1, EU 2.1, EU 2.2, EU 2.3, EU 5.2, EU 5.3, EU 5.5, EU 6.1

Projects and Major Assignments:

- Designing an Adder in Logicly
- Designing Memory Storage in Logicly
- Programming a Simulation with Numbers or Dice
- Programming a Real-World Simulation

Guiding Questions

- **How can you convert numbers between binary, decimal, and hexadecimal form?**
- **What are some uses for binary and hexadecimal representations of numbers?**
- **How do computers add numbers, and how do they store numbers?**
- **How is a computer's random access memory (RAM) and central processing unit (CPU) organized?**
- **How do random numbers allow computers to simulate real-world events?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 2.1.1: Describe the variety of abstractions used to represent data. [P3]	**Web** "Cisco Binary Game"	**Instructional Activity: Binary Representation of Data** Students look up the definition of *bit* from two different sources, learn different ways to physically represent a bit (e.g., lights, punch cards, magnetic polarity), and use a table to translate letters into ASCII binary. The next day they learn to convert between binary and decimal, and they play an online game to practice their conversion skills. Using an online converter, they see how binary representations of numbers such as 0.8 lead to rounding errors. **Essential knowledge addressed: 2.1.1** A, B, E
LO 2.1.1: Describe the variety of abstractions used to represent data. [P3] **LO 2.1.2:** Explain how binary sequences are used to represent digital data. [P5]	**Web** Excerpt from James Glieck's *The Information*	**Instructional Activity: The Information** Students read the excerpt, in which the author uses information theory to explain physics, economics, genetics, and other areas of study, and they learn to see bits as something all encompassing. It is a great way to show students the broad scope of this course. **Essential knowledge addressed: 2.1.1**C, **2.1.2**F
LO 5.2.1: Explain how programs implement algorithms. [P3]	**Web** "Hamming Code (7,4) Introduction to Telecom-munication"	**Instructional Activity: Self-Correcting Codes** Students learn to spot transmission errors by performing parity checks. They check for errors in a single string of data and then learn to correct errors in a two-dimensional grid of bits. Finally, students watch the video and learn the algorithm for hamming codes, which are a kind of self-correcting binary code. **Essential knowledge addressed: 5.2.1**J

This article has a high reading level. In such cases I like to have volunteers read a paragraph out loud. After each paragraph, we use our district reading strategies to process the text.

While hamming codes are not part of the curricular framework, it is useful for students to learn them as a way to see other uses for binary representations of data. Learning hamming codes also provides good practice in following an algorithm.

UNIT 4: ABSTRACTION AND SIMULATION

Estimated Time: 21 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 5 Programming
BIG IDEA 6 The Internet

Essential Understandings:

- EU 1.1, EU 2.1, EU 2.2, EU 2.3, EU 5.2, EU 5.3, EU 5.5, EU 6.1

Projects and Major Assignments:

- Designing an Adder in Logicly
- Designing Memory Storage in Logicly
- Programming a Simulation with Numbers or Dice
- Programming a Real-World Simulation

Guiding Questions

- **How can you convert numbers between binary, decimal, and hexadecimal form?**
- **What are some uses for binary and hexadecimal representations of numbers?**
- **How do computers add numbers, and how do they store numbers?**
- **How is a computer's random access memory (RAM) and central processing unit (CPU) organized?**
- **How do random numbers allow computers to simulate real-world events?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 2.2.1: Develop an abstraction when writing a program or creating other computational artifacts. [P2] **LO 5.3.1:** Use abstraction to manage complexity in programs. [P3]	**Web** "Abstraction and Testing"	**Instructional Activity: Introduction to Snap!** Abstraction is the theme of this unit, and in this lesson, students learn that abstraction can refer to software as well as hardware. The Snap! programming language allows users to create their own blocks (procedures) that take parameters and return values. Students go through a series of examples to make blocks that draw a square of a certain size, draw a regular polygon, and report back various numbers, words, or Boolean values. **Essential knowledge addressed: 2.2.1** A-C, **5.3.1** A-G
LO 2.1.1: Describe the variety of abstractions used to represent data. [P3] **LO 6.1.1:** Explain the abstractions in the Internet and how the Internet functions. [P3]		**Instructional Activity: Hexadecimal Numbers** In this activity, students learn to convert between hexadecimal, binary, and decimal numbers in three ways. First, they count from 1 to 16 using hex. Next, they match each hexadecimal digit they wrote with its equivalent in binary. Finally, they learn to use place value to convert hex to decimal. Once students have practiced these skills, they do an Internet search of IPv6 addresses. On a piece of paper, they write down two different IPv6 addresses and write their equivalents in binary. **Essential knowledge addressed: 2.1.1** D-G; **6.1.1** G, H
LO 5.5.1: Employ appropriate mathematical and logical concepts in programming. [P1]		**Instructional Activity: Review of AND, OR, and NOT Using Snap!** In anticipation of the next activity on logic gates AND, OR, and NOT, students clarify their usage by practicing writing Snap! reporter blocks that use these ideas. **Essential knowledge addressed: 5.5.1** E-G

I like to show my students places where they have already seen hexadecimal numbers, such as MAC addresses, HTML colors, and Unicode characters.

UNIT 4: ABSTRACTION AND SIMULATION

Estimated Time: 21 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 5 Programming
BIG IDEA 6 The Internet

Essential Understandings:

- EU 1.1, EU 2.1, EU 2.2, EU 2.3, EU 5.2, EU 5.3, EU 5.5, EU 6.1

Projects and Major Assignments:

- Designing an Adder in Logicly
- Designing Memory Storage in Logicly
- Programming a Simulation with Numbers or Dice
- Programming a Real-World Simulation

Guiding Questions

- **How can you convert numbers between binary, decimal, and hexadecimal form?**
- **What are some uses for binary and hexadecimal representations of numbers?**
- **How do computers add numbers, and how do they store numbers?**
- **How is a computer's random access memory (RAM) and central processing unit (CPU) organized?**
- **How do random numbers allow computers to simulate real-world events?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 2.2.3: Identify multiple levels of abstractions that are used when writing programs. [P3]	**Web** Logicly	**Instructional Activity: Logic Gates and Truth Tables** Students are introduced to the logic.ly website and they use the free Web-based demo module. This site is a wonderful way for students to experiment with logic gates in a way that allows them to construct their own knowledge. After a little bit of playing, students write tables of values for AND, OR, NOT, and XOR. Students then make more complicated gates and use truth tables to list the output for all combinations of inputs. **Essential knowledge addressed: 2.2.3** D-F	*Many of the instructional activities that use Logicly are not part of the curriculum framework, but I find such activities useful to give students a richer view of computer science in a fun, hands-on way.*
LO 2.1.2: Explain how binary sequences are used to represent digital data. [P5] **LO 2.2.3:** Identify multiple levels of abstractions that are used when writing programs. [P3]	**Web** Logicly	**Instructional Activity: Designing Adders in Logicly** In this activity, students use logic gates to demonstrate how computers add two bits. Logicly allows students to abstract a complicated circuit into a "box" with inputs and outputs. Once students create the box for an adder with carry-in and carry-out bits, they can build a circuit that adds two 4-bit numbers. Students naturally see a problem when some results are not correct, and this observation leads to a student-constructed conversation about overflow error. **Essential knowledge addressed: 2.1.2**B, **2.2.3** D-F	*The free Web-based version of Logicly does not allow students to save or open files, so students take screen shots of their work. I have a paid version on my home computer, which I use to store my files.*
LO 2.1.1: Describe the variety of abstractions used to represent data. [P3]		**Formative Assessment: Quiz** Students take a short quiz on how to convert between binary, decimal, and hexadecimal representations. The quiz has three columns: "decimal," "binary," and "hex." Each row has a number in one base, and the students need to fill in the missing cells in each row. **Essential knowledge addressed: 2.1.1**G	*I return the quizzes to students and ask them to make corrections, showing the steps involved in the calculations. I provide feedback by conversing with each student to verify that his or her corrections are accurate.*

UNIT 4: ABSTRACTION AND SIMULATION

Estimated Time: 21 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 5 Programming
BIG IDEA 6 The Internet

Essential Understandings:

- EU 1.1, EU 2.1, EU 2.2, EU 2.3, EU 5.2, EU 5.3, EU 5.5, EU 6.1

Projects and Major Assignments:

- Designing an Adder in Logicly
- Designing Memory Storage in Logicly
- Programming a Simulation with Numbers or Dice
- Programming a Real-World Simulation

Guiding Questions

- **How can you convert numbers between binary, decimal, and hexadecimal form?**
- **What are some uses for binary and hexadecimal representations of numbers?**
- **How do computers add numbers, and how do they store numbers?**
- **How is a computer's random access memory (RAM) and central processing unit (CPU) organized?**
- **How do random numbers allow computers to simulate real-world events?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 2.2.3: Identify multiple levels of abstractions that are used when writing programs. [P3]		**Instructional Activity: Storing a Bit in Logicly** At a most basic level, logic gates help students understand how computers can store data in random access memory. Using Logicly, students build their own "latches" for storing one bit, abstract that into a box with inputs and outputs, and then create multiple memory addresses. This helps students understand why their computers and gaming consoles are described as 32 bit or 64 bit. **Essential knowledge addressed: 2.2.3** D-F
LO 2.1.2: Explain how binary sequences are used to represent digital data. [P5] **LO 2.2.3:** Identify multiple levels of abstractions that are used when writing programs. [P3] **LO 5.2.1:** Explain how programs implement algorithms. [P3]	**Web** "Online CS Modules: The Central Processing Unit"	**Instructional Activity: Simulating Operation of a CPU** Virginia Tech's interactive website connects binary numbers to an understanding of what is happening inside a computer's central processing unit (CPU). Students interact with the site to learn how different combinations of binary digits can represent instructions or data. They see how programs are executed at a very low level on a computer. **Essential knowledge addressed: 2.1.2** D-F, **2.2.3** G-I, **5.2.1**F
LO 2.3.2: Use models and simulations to formulate, refine, and test hypotheses. [P3]	**Web** "Animations" Conway's Game of Life	**Instructional Activity: Simulations** Students use a website to investigate Conway's Game of Life, one of the original computer simulations. After viewing some engineering simulations, students hold a class discussion about how simulations are helpful in scientific and other applications. Students do Web searches for additional simulations. Each student shares with the class one online simulation that he or she found. **Essential knowledge addressed: 2.3.2** A-H

Knowledge of the operation of a CPU is not directly required in the curriculum framework, but it helps students see why binary numbers are important in computer science.

Conway's Game of Life can get really addicting. I spend an entire class period on this classic simulation.

UNIT 4: ABSTRACTION AND SIMULATION

Estimated Time: 21 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 5 Programming
BIG IDEA 6 The Internet

Essential Understandings:

- EU 1.1, EU 2.1, EU 2.2, EU 2.3, EU 5.2, EU 5.3, EU 5.5, EU 6.1

Projects and Major Assignments:

- Designing an Adder in Logicly ▶ Designing Memory Storage in Logicly ▶ Programming a Simulation with Numbers or Dice ▶ Programming a Real-World Simulation

Guiding Questions

▶ **How can you convert numbers between binary, decimal, and hexadecimal form?** ▶ **What are some uses for binary and hexadecimal representations of numbers?** ▶ **How do computers add numbers, and how do they store numbers?** ▶ **How is a computer's random access memory (RAM) and central processing unit (CPU) organized?** ▶ **How do random numbers allow computers to simulate real-world events?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 2.3.1: Use models and simulations to represent phenomena. [P3] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]		**Instructional Activity: Programming Basic Simulations with Discrete Events** Following my lead, students make short programs that do the following: ▶ Simulate rolling two dice ▶ Simulate rolling two dice until "doubles" occurs ▶ Simulate tossing a coin 100 times and recording the number of heads The next day in class, students write a program that simulates the cereal box problem. In this problem, a company that manufactures cereal randomly places one of six tokens in each box. If a person collects all six tokens, he or she wins a prize. How many boxes of cereal would a person need to purchase to collect all six tokens? **Essential knowledge addressed: 2.3.1** B, C; **5.5.1** A, D, J
LO 2.3.1: Use models and simulations to represent phenomena. [P3] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]	**Web** "Retirement Nest Egg Calculator"	**Instructional Activity: Programming Monte Carlo Simulations** Students learn about Monte Carlo simulations by visiting a retirement-planning website. As a class, they program their own Monte Carlo simulation that approximates the value of pi. **Essential knowledge addressed: 2.3.1**B; **5.5.1** A, D, J

In this activity, I model how a programmer approaches a problem, ask different students to take over the program, and show common errors and how to fix them.

UNIT 4: ABSTRACTION AND SIMULATION

Estimated Time: 21 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 5 Programming
BIG IDEA 6 The Internet

Essential Understandings:
▶ EU 1.1, EU 2.1, EU 2.2, EU 2.3, EU 5.2, EU 5.3, EU 5.5, EU 6.1

Projects and Major Assignments:
▶ Designing an Adder in Logicly ▶ Designing Memory Storage in Logicly ▶ Programming a Simulation with Numbers or Dice ▶ Programming a Real-World Simulation

Guiding Questions

▶ **How can you convert numbers between binary, decimal, and hexadecimal form?** ▶ **What are some uses for binary and hexadecimal representations of numbers?** ▶ **How do computers add numbers, and how do they store numbers?** ▶ **How is a computer's random access memory (RAM) and central processing unit (CPU) organized?** ▶ **How do random numbers allow computers to simulate real-world events?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 2.3.1:** Use models and simulations to represent phenomena. [P3] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]		**Instructional Activity: Simulating Real-World Events** Students write their own real-world simulation. They may choose any topic they wish, but they must use random numbers. **Essential knowledge addressed: 1.1.1**B; **2.3.1** A-D; **5.5.1** A, D, J	*Students should be encouraged to choose their own idea, but some examples might include simulating sporting events, elections, permutations, the stock market, or the chance meeting of friends in a group.*
All of the learning objectives in this unit are addressed.		**Summative Assessment: Unit Quiz** Students take a quiz consisting of approximately 10 short-answer questions (e.g., *What is a Monte Carlo simulation?* and *Why are procedures useful?*) and 10 coding questions, which ask students to repeat tasks that are similar to pieces of the programs they wrote in this unit. **All of the unit's essential knowledge statements are addressed.**	*This summative assessment addresses all of the guiding questions for this unit.*

UNIT 5: DATA

Estimated Time: 20 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.2, EU 2.2, EU 3.1, EU 3.2, EU 3.3, EU 4.1, EU 4.2, EU 5.1, EU 5.3, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3

Projects and Major Assignments:

- Designing a School Data Wall
- You-Sort Activity
- Programming a Reverse Guessing Game
- Coding a Bioinformatics Algorithm
- Analyzing Data — Gapminder.org

Guiding Questions

- **How does continuous access to large amounts of data change how people and organizations make decisions?**
- **How do computers put things in order and find things in a list?**
- **What is the connection between data, information, knowledge, and wisdom?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.2.5: Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **LO 3.2.2:** Determine how large data sets impact the use of computational processes to discover information and knowledge. [P3] **LO 7.2.1:** Explain how computing has impacted innovations in other fields. [P1]	**Web** "Everything You've Always Wanted to Know About Akinator"	**Instructional Activity: Innovations from Data** The last 20 years have seen waves of trends in computing. Whether it was hardware, software, Internet, search, social, or mobile, each wave created incredible consumer innovations, as well as profits for companies that created those innovations. Will data be the next wave? Students consider all this as they play an online guessing game that is powered by crowd-sourced data, analyze the game, and contribute to the game's data. **Essential knowledge addressed: 1.2.5** A-D; **3.2.2**A; **7.2.1** C, D
LO 3.2.2: Determine how large data sets impact the use of computational processes to discover information and knowledge. [P3] **LO 7.1.1:** Explain how computing innovations affect communication, interaction, and cognition. [P4] **LO 7.2.1:** Explain how computing has impacted innovations in other fields. [P1]	**Web** "Connected Cars. Connected Homes. Connected Lives"	**Instructional Activity: Big Data** Students use Wikipedia to learn how to quantify large amounts of data such as gigabyte, terabyte, and petabyte. Students then watch a video about big data and discuss how big data changes how people and organizations will make decisions. **Essential knowledge addressed: 3.2.2** A, F; **7.1.1** J, K; **7.2.1**A

The AP Exam does not test knowledge of units of data such as petabyte, but this knowledge is practical and transferable to other areas of students' academic and personal lives.

UNIT 5: DATA

Estimated Time: 20 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.2, EU 2.2, EU 3.1, EU 3.2, EU 3.3, EU 4.1, EU 4.2, EU 5.1, EU 5.3, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3

Projects and Major Assignments:

- Designing a School Data Wall
- You-Sort Activity
- Programming a Reverse Guessing Game
- Coding a Bioinformatics Algorithm
- Analyzing Data — Gapminder.org

Guiding Questions

- **How does continuous access to large amounts of data change how people and organizations make decisions?**
- **How do computers put things in order and find things in a list?**
- **What is the connection between data, information, knowledge, and wisdom?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 3.2.1: Extract information from data to discover and explain connections or trends. [P1] **LO 3.2.2:** Determine how large data sets impact the use of computational processes to discover information and knowledge. [P3] **LO 7.1.1:** Explain how computing innovations affect communication, interaction, and cognition. [P4]	**Web** "Google Flu Trends" "Google Ngram Viewer" "THINK"	**Instructional Activity: Visualizing Big Data** Students read about how organizations are using data visualizations to make decisions. Students read about data walls and how they are used to visualize real-time data. Students write a paragraph explaining where in their school they would place a data wall, what information it would contain, and how it would be visualized. Students visit websites that allow them to look at large amounts of data, such as Google Ngrams and Google Flu Trends. **Essential knowledge addressed: 3.2.1** A-E; **3.2.2** A, B; **7.1.1** G, J, K	*Asking students to design possible computational artifacts helps them think creatively, as well as practice their writing.*
LO 4.1.1: Develop an algorithm for implementation in a program. [P2] **LO 4.1.2:** Express an algorithm in a language. [P5] **LO 4.2.4:** Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity. [P4]		**Instructional Activity: You-Sort** I give pairs of students small pieces of paper with integers on them, turned upside down. One student is allowed to look at the numbers and tell the other student which of two numbers is larger. The other student is allowed to choose two numbers and rearrange the slips of paper. The goal is to have the second student find a way to place all the numbers in order without ever looking at the numbers. On paper, students write down the steps in their sorting algorithm. **Essential knowledge addressed: 4.1.1** G-I; **4.1.2** A, B, G, I; **4.2.4** F, G	*It is important for students to devise their own algorithms for solving a problem before they learn a known algorithm. The knowledge that students construct in this activity serves as a strong foundation for many intermediate computer science topics.*

UNIT 5: DATA

Estimated Time: 20 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

▶ EU 1.2, EU 2.2, EU 3.1, EU 3.2, EU 3.3, EU 4.1, EU 4.2, EU 5.1, EU 5.3, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3

Projects and Major Assignments:

▶ Designing a School Data Wall ▶ You-Sort Activity ▶ Programming a Reverse Guessing Game ▶ Coding a Bioinformatics Algorithm ▶ Analyzing Data — Gapminder.org

Guiding Questions

▶ **How does continuous access to large amounts of data change how people and organizations make decisions?**
▶ **How do computers put things in order and find things in a list?** ▶ **What is the connection between data, information, knowledge, and wisdom?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 4.1.1: Develop an algorithm for implementation in a program. [P2] **LO 4.2.1:** Explain the difference between algorithms that run in a reasonable time and those that do not run in a reasonable time. [P1] **LO 4.2.4:** Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity. [P4]		**Instructional Activity: Sorting Algorithms** Students visit the Wikipedia pages that describe selection sort and insertion sort, and then they act out each algorithm using paper and pencil to move numbers around in a list. Using their writing from the day before, students determine if their sort was close to one of these three sorts. **Essential knowledge addressed: 4.1.1** E, G, H; **4.2.1** A, B; **4.2.4** D-G
LO 4.1.2: Express an algorithm in a language. [P5] **LO 5.4.1:** Evaluate the correctness of a program. [P4] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]		**Instructional Activity: Programming Bubble Sort** As a class, students discuss the algorithm for bubble sort while coding it in Python. Next, students code other sorts, either by their own design or by looking up the code on the Internet. Finally, students use the `sort()` function built into Python to sort a list. **Essential knowledge addressed: 4.1.2** C, I; **5.4.1** C, E, I, J, N; **5.5.1** G, H
LO 4.2.4: Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity. [P4] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]		**Instructional Activity: Binary Search Worksheet** Students construct knowledge of the binary search algorithm by playing a guessing game with a partner. In the game, one student thinks of a number between 1 and 64. The other student tries to guess the number. The partner who is keeping the number may tell his or her partner if the guess was "too high," "too low," or "correct." Students analyze this algorithm and compare it with a linear search. The next day, I lead students in writing a Python function that implements a linear search and another that implements a binary search. **Essential knowledge addressed: 4.2.4** A, B, D-H; **5.5.1** A, J

Knowing particular sorting algorithms is not in the curriculum framework, but I use these algorithms to have students practice acting out algorithms by hand. This also allows students to ask rich questions (e.g., "Which sort is the best?"), which they can then choose to investigate on their own.

UNIT 5: DATA

Estimated Time: 20 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

▶ EU 1.2, EU 2.2, EU 3.1, EU 3.2, EU 3.3, EU 4.1, EU 4.2, EU 5.1, EU 5.3, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3

Projects and Major Assignments:

▶ Designing a School Data Wall ▶ You-Sort Activity ▶ Programming a Reverse Guessing Game ▶ Coding a Bioinformatics Algorithm ▶ Analyzing Data — Gapminder.org

Guiding Questions

▶ **How does continuous access to large amounts of data change how people and organizations make decisions?**
▶ **How do computers put things in order and find things in a list?** ▶ **What is the connection between data, information, knowledge, and wisdom?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 4.1.2: Express an algorithm in a language. [P5] **LO 5.1.2:** Develop a correct program to solve problems. [P2] **LO 5.1.3:** Collaborate to develop a program. [P6] **LO 5.3.1:** Use abstraction to manage complexity in programs. [P3]		**Instructional Activity: Programming a Reverse Guessing Game** Rather than coding a binary search, students write a program in Python that plays a reverse guessing game. In this game, the student thinks of an integer, and the program guesses the number. The class as a group writes the code, incorporating different ideas from individual students. Students import the Python random module for the first time and discuss how and why modules are imported. **Essential knowledge addressed: 4.1.2**C; **5.1.2** A-E; **5.1.3** B, D; **5.3.1** M-0
LO 5.3.1: Use abstraction to manage complexity in programs. [P3] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1] **LO 7.2.1:** Explain how computing has impacted innovations in other fields. [P1]	**Web** "Locations"	**Instructional Activity: Bioinformatics Algorithms** Using the context of bioinformatics, students code list algorithms in Python. As an extension, students visit a website that offers more challenging problems in bioinformatics. **Essential knowledge addressed: 5.3.1** M-0; **5.5.1** H-J; **7.2.1** A, B, E

I start this lesson by asking students if any of them have heard of the term bioinformatics. *I know that it is discussed in our freshman biology course. In case no one remembers, I remind (or a student reminds) the class how the information stored in DNA can be represented by the letters A, C, T, and G.*

UNIT 5: DATA

Estimated Time: 20 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.2, EU 2.2, EU 3.1, EU 3.2, EU 3.3, EU 4.1, EU 4.2, EU 5.1, EU 5.3, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3

Projects and Major Assignments:

- Designing a School Data Wall
- You-Sort Activity
- Programming a Reverse Guessing Game
- Coding a Bioinformatics Algorithm
- Analyzing Data — Gapminder.org

Guiding Questions

- **How does continuous access to large amounts of data change how people and organizations make decisions?**
- **How do computers put things in order and find things in a list?**
- **What is the connection between data, information, knowledge, and wisdom?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 3.3.1: Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information. [P4]	**Print** MacCormack, chapter 7 **Web** "Computer Science Principles Widgets"	**Instructional Activity: Lossless Compression** Lossless compression converts a file into another file that preserves all information from the original. I provide students this definition, and then they look up the definition for themselves on the Internet and compare the two. Then, as a class, they generate examples of this kind of compression. They act out the run length encoding algorithm on paper and use an Internet widget to demonstrate substitution compression. **Essential knowledge addressed: 3.3.1** C, D	
LO 3.3.1: Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information. [P4]	**Web** "Online Photo Editor"	**Instructional Activity: Lossy Compression** Lossy compression converts a file into another file that does not preserve all information from the original. I provide students this definition, and then they look up the definition for themselves on the Internet and compare the two. Then, as a class, they generate examples of uses of this kind of compression. They create pixilated pictures using an online photo editor website. **Essential knowledge addressed: 3.3.1** C, E	
LO 3.3.1: Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information. [P4] **LO 5.4.1:** Evaluate the correctness of a program. [P4]		**Formative Assessment: Coding Data Compression** Students individually write Python programs that use lossy compression. One of their programs should act like a texting app and remove all vowels from a message. Another should take a list of numbers and create a new list with every other number in it. **Essential knowledge addressed: 3.3.1** C-E, **5.4.1** A-G	*Students show me their codes and test them thoroughly. I give them feedback on the accuracy of their work, as well as the style of their code.*

UNIT 5: DATA

Estimated Time: 20 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.2, EU 2.2, EU 3.1, EU 3.2, EU 3.3, EU 4.1, EU 4.2, EU 5.1, EU 5.3, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3

Projects and Major Assignments:

- Designing a School Data Wall ▶ You-Sort Activity ▶ Programming a Reverse Guessing Game ▶ Coding a Bioinformatics Algorithm ▶ Analyzing Data — Gapminder.org

Guiding Questions

▶ How does continuous access to large amounts of data change how people and organizations make decisions?

▶ How do computers put things in order and find things in a list? ▶ What is the connection between data, information, knowledge, and wisdom?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 3.1.1: Find patterns and test hypotheses about digitally processed information to gain insight and knowledge. [P4]	**Web** Topaz, "The Hitchhiker's Guide to Nursing Informatics Theory: Using the Data-Knowledge-Information-Wisdom Framework to Guide Informatics Research"	**Instructional Activity: The DIKW Pyramid** The data-information-knowledge-wisdom (DIKW) pyramid is used in many fields to explain how businesses and other organizations will make data-related decisions. During a lecture, students take active notes on this hierarchy, consider an example from the Web, and then make up their own examples in an area that they are passionate about. **Essential knowledge addressed: 3.1.1** A, D, E
LO 3.1.1: Find patterns and test hypotheses about digitally processed information to gain insight and knowledge. [P4] **LO 3.1.2:** Collaborate when processing information to gain insight and knowledge. [P6] **LO 3.1.3:** Explain the insight and knowledge gained from digitally processed data by using appropriate visualizations, notations, and precise language. [P5]	**Web** "200 Countries, 200 Years, 4 Minutes" "Wealth and Health of Nations"	**Instructional Activity: Gapminder** Students watch a video by Hans Rosling, the creator of the Gapminder website. Rosling uses data about the health of nations to correct common misconceptions that people have about developing countries. After watching the video, working with a partner, students use the interactive "Wealth and Health of Nations" to find their own patterns in the data. Finally, they use their creativity to imagine a similar website that they could create about an area they are passionate about. **Essential knowledge addressed: 3.1.1** B, C, E; **3.1.2** A-F; **3.1.3** A-E

UNIT 5: DATA

Estimated Time: 20 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

▶ EU 1.2, EU 2.2, EU 3.1, EU 3.2, EU 3.3, EU 4.1, EU 4.2, EU 5.1, EU 5.3, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3

Projects and Major Assignments:

▶ Designing a School Data Wall ▶ You-Sort Activity ▶ Programming a Reverse Guessing Game ▶ Coding a Bioinformatics Algorithm ▶ Analyzing Data — Gapminder.org

Guiding Questions

▶ **How does continuous access to large amounts of data change how people and organizations make decisions?**
▶ **How do computers put things in order and find things in a list?** ▶ **What is the connection between data, information, knowledge, and wisdom?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 3.2.2: Determine how large data sets impact the use of computational processes to discover information and knowledge. [P3] **LO 3.3.1:** Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information. [P4]		**Instructional Activity: Data Use in Your School** I invite an administrator from the school district to talk to the class about data use. Students ask the administrator questions about how the school collects data, what data it collects, how it is stored, how it is visualized, and what decision are made about the data. Students also make suggestions to the administrator about what data they think is important to track in order to lead to knowledge and wisdom. Students write a two-paragraph summary of the talk. **Essential knowledge addressed: 3.2.2** B-H; **3.3.1** A, B, F-I	*Students may not realize how much data their schools are using. This discussion could also involve an IT director who could discuss how the data is stored, maintained, and kept private.*
LO 3.2.1: Extract information from data to discover and explain connections or trends. [P1] **LO 3.2.2:** Determine how large data sets impact the use of computational processes to discover information and knowledge. [P3] **LO 3.3.1:** Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information. [P4] **LO 7.3.1:** Analyze the beneficial and harmful effects of computing. [P4]		**Instructional Activity: Privacy in the Age of Big Data** Students read articles and watch videos about the ability of governments, businesses, and individuals to maintain information about citizens. I explain metadata and as a class we discuss connections to privacy. Students consider current issues in data privacy and write an essay stating their position on what laws, if any, should enforce data privacy. **Essential knowledge addressed: 3.2.1** G-I; **3.2.2**D; **3.3.1**F; **7.3.1** G, H, J-M	*There are many possible articles and videos. Choose something that is current and interests your students.*

UNIT 5: DATA

Estimated Time: 20 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

▶ EU 1.2, EU 2.2, EU 3.1, EU 3.2, EU 3.3, EU 4.1, EU 4.2, EU 5.1, EU 5.3, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3

Projects and Major Assignments:

▶ Designing a School Data Wall ▶ You-Sort Activity ▶ Programming a Reverse Guessing Game ▶ Coding a Bioinformatics Algorithm ▶ Analyzing Data — Gapminder.org

Guiding Questions

▶ **How does continuous access to large amounts of data change how people and organizations make decisions?**
▶ **How do computers put things in order and find things in a list?** ▶ **What is the connection between data, information, knowledge, and wisdom?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 3.1.2: Collaborate when processing information to gain insight and knowledge. [P6] **LO 3.1.3**: Explain the insight and knowledge gained from digitally processed data by using appropriate visualizations, notations, and precise language. [P5] **LO 3.2.1**: Extract information from data to discover and explain connections or trends. [P1]		**Instructional Activity: Downloading Public Data Sets into Spreadsheets** Working with a partner, students search for public data sets (from a local, state, or federal website) that match their interests. Through online research, they learn what comma-separated values (CSV) files are and how they are formatted. Finally, students download the public data sets they found and open them using a spreadsheet. **Essential knowledge addressed: 3.1.2** A-F; **3.1.3** A-E; **3.2.1** D, F	*Your city or state government may have a website of public data. Local data is always of interest to students.* *If Excel cannot open a CSV file automatically, use the dialog box after choosing "Open" to select an option that includes "CSV." In Google Sheets, first upload the file to Google Drive, then select the file, and choose "export to spreadsheet."*
LO 2.2.2: Use multiple levels of abstraction to write programs. [P3] **LO 3.1.3**: Explain the insight and knowledge gained from digitally processed data by using appropriate visualizations, notations, and precise language. [P5] **LO 5.3.1**: Use abstraction to manage complexity in programs. [P3]	**Web** "Thumbnail Gallery"	**Instructional Activity: Manipulating Data in Python** Students open data files in Python, store them in lists or dictionaries, and use Python's matplotlib library to visualize the data. **Essential knowledge addressed: 2.2.2**A, **3.1.3** A-D, **5.3.1** M-0	*Even if your teaching situation does not allow you to import this library, your class can visit the website to see the examples and look at the application programming interface (API).*

UNIT 5: DATA

Estimated Time: 20 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

▶ EU 1.2, EU 2.2, EU 3.1, EU 3.2, EU 3.3, EU 4.1, EU 4.2, EU 5.1, EU 5.3, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3

Projects and Major Assignments:

▶ Designing a School Data Wall ▶ You-Sort Activity ▶ Programming a Reverse Guessing Game ▶ Coding a Bioinformatics Algorithm ▶ Analyzing Data — Gapminder.org

Guiding Questions

▶ **How does continuous access to large amounts of data change how people and organizations make decisions?**
▶ **How do computers put things in order and find things in a list?** ▶ **What is the connection between data, information, knowledge, and wisdom?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives in this unit are addressed.		**Summative Assessment: Unit Test** Students take an open-ended test, with approximately 20 short and medium-length questions on topics such as sizes of big data, the DIKW pyramid, Gapminder, using Python to manipulate data, CSV files, compression, and bioinformatics; for example, *List what the letters in the DIKW pyramid stand for. Then use a specific data set from the Gapminder website to explain each term.* **All of the unit's essential knowledge statements are addressed.**

This summative assessment addresses all of the guiding questions for this unit.

UNIT 6: INTRACTABLE PROBLEMS

Estimated Time: 8 Hours

BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 6 The Internet

Essential Understandings:
- EU 4.2, EU 5.3, EU 6.3

Projects and Major Assignments:
- Creating Secret Messages
- Programming a Caesar Cipher

Guiding Questions

- What kinds of problems are hard for computers to solve?
- What kinds of problems are impossible for computers to solve?
- How do hard problems form the basis for modern encryption?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 4.2.1: Explain the difference between algorithms that run in a reasonable time and those that do not run in a reasonable time. [P1]	**Web** "The Traveling Salesman with Simulated Annealing, R, and Shiny"	**Instructional Activity: Intractable Problems and Heuristics** Students solve examples of the traveling salesperson problem, largest-clique problem, knapsack problem, and three-coloring problem by hand, and they visit a website that demonstrates the traveling salesman problem. Students discuss brute force methods and compute the number of possibilities a brute force solution must try to solve a problem of size *n*. **Essential knowledge addressed: 4.2.1** A-D	*These problems are fun to solve, both by hand and with applets. The work students do solving the problems by hand helps them understand heuristics.*
LO 6.3.1: Identify existing cybersecurity concerns and potential options to address these issues with the Internet and the systems built on it. [P1]	**Web** "158,962,555, 217,826,360,000 (Enigma Machine)" "Codes, Ciphers, Encryption and Cryptography" "Enigma Simulation"	**Instructional Activity: Cryptography** Students play a game called the Encryption Game. In this game, students work in pairs to create a system to encode a two-digit number. One student sends a message across the room so the entire class can see and hear it. The other students try to guess the encrypted number. Finally, the partner decodes the number in front of the class. The next day students practice some basic cryptographic algorithms such as the "Caesar Cipher" and the "One-time Pad" from "Codes, Ciphers, Encryption and Cryptography." Students send messages to each other using the class online management system (Moodle). Students watch a video about the Enigma machine and send each other messages using an online simulator. **Essential knowledge addressed: 6.3.1** H-K	*The Encryption Game allows students to show creativity and is always a well-liked activity. Students like to send each other secret messages, even when they include the key. Make sure to remind students that they need to send appropriate messages that are not offensive.*
LO 5.3.1: Use abstraction to manage complexity in programs. [P3]		**Instructional Activity: Programming a Caesar Cipher** Students use the "Unicode" and "Unicode as letter" blocks in Snap! to write a program that implements a Caesar cipher. **Essential knowledge addressed: 5.3.1** I, M-0	

UNIT 6: INTRACTABLE PROBLEMS

Estimated Time: 8 Hours

BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 6 The Internet

Essential Understandings:
- EU 4.2, EU 5.3, EU 6.3

Projects and Major Assignments:
- Creating Secret Messages
- Programming a Caesar Cipher

Guiding Questions

▶ What kinds of problems are hard for computers to solve? ▶ What kinds of problems are impossible for computers to solve? ▶ How do hard problems form the basis for modern encryption?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 6.3.1: Identify existing cybersecurity concerns and potential options to address these issues with the Internet and the systems built on it. [P1]	**Web** "Cryptograms" "Encryption and HUGE Numbers" "Simple RSA Key Generation"	**Instructional Activity: Public-Key Encryption** Students watch a video about public-key encryption, do a simple calculation by hand, and use a Web applet that implements RSA encryption. Students send each other messages using an online RSA encoder/decoder. Students learn that RSA encryption can be broken, but not in a reasonable time. **Essential knowledge addressed: 6.3.1** J, L
LO 6.3.1: Identify existing cybersecurity concerns and potential options to address these issues with the Internet and the systems built on it. [P1]	**Web** "Network Solutions SSL Certificate"	**Instructional Activity: Certificate Authorities** Students learn how certificate authorities (CAs) work by researching the subject on the Internet, and then they discuss as a class how CAs relate to public-key encryption and to their knowledge of the Internet. **Essential knowledge addressed: 6.3.1** A, K
LO 4.2.2: Explain the difference between solvable and unsolvable problems in computer science. [P1] **LO 4.2.3:** Explain the existence of undecidable problems in computer science. [P1]	**Print** MacCormick, chapter 10	**Instructional Activity: Unsolvable Problems** Students begin by writing Python loops that terminate. Then they write Python loops that do not terminate. One unsolvable problem in computer science is to write a program that determines whether another program comes to a halt or not. Students research the definition of a *halting problem* on the Internet and share their search results out loud with the class. **Essential knowledge addressed: 4.2.2**D, **4.2.3** A-C
All of the learning objectives in this unit are addressed.		**Summative Assessment: Unit Quiz** Students take a quiz on the topics of unsolvable problems, CAs, public-key encryption, Enigma, ciphers, and different kinds of intractable problems. There are approximately 20 short-answer questions on the quiz. **All of the unit's essential knowledge statements are addressed.**

The resource for this activity is written at a teacher level, but it may be of interest to some students.

This summative assessment addresses all of the guiding questions for this unit.

AP Performance Task

ESTIMATED TIME:
12 hours

Create – Applications from Ideas

At this point students are ready to complete the Create Performance Task. They have had numerous experiences writing complicated programs in a variety of contexts. Students have written programs in the Scratch, Snap!, and Python languages, and they choose one of these languages for their programs. (I have not had a student choose another language, but students could if they wanted to. Some students have chosen to use Scratch, even though we have spent the least amount of time coding in this language. I have asked these students why they selected Scratch, and their responses indicate that they had used Scratch in a previous course or in middle school, and for that reason they are more comfortable with it.)

Students have had chances to work with partners in many different ways in the course. I encourage them to consider their experience working in pairs and prepare well ahead of time in choosing their partners for the performance task to avoid making a poor choice or having students lose out due to absences on any given day.

Some teachers like to have students do a practice performance task before they complete the official one to be submitted to the College Board. This is a good idea for students if your curriculum can accommodate it. For my classes, I like to use the first-semester final as a mini Create Performance Task. Students do the collaborative program and write up the collaborative responses.

Before students begin the performance task, I go through our school's reading protocol (a prescribed way to read a complicated text) with them and make sure that they have no questions. I let students know ahead of time how many class periods we will devote to the performance task to ensure that we meet the number of hours specified in the directions.

I try to keep a cheerful and productive atmosphere in class during this time so that the students can produce their best work.

UNIT 7: IMPACT ON SOCIETY

Estimated Time: 8 Hours

BIG IDEA 7 Global Impact

Essential Understandings:
- EU 7.1, EU 7.3, EU 7.4, EU 7.5

Projects and Major Assignments:
- Copyright and the Law Paper
- Class Debates

Guiding Questions

- How do we define the impact of computing innovations?
- What are the beneficial or harmful effects of various computing innovations?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 7.4.1: Explain the connections between computing and real-world contexts, including economic, social, and cultural contexts. [P1]		**Formative Assessment: Global Impact** Students discuss the following questions: *What does it mean to have a large impact? What kinds of past innovations have had the most impact?* I guide students through some previous global innovations, such as the telephone. For each impact discussed, students individually fill out a worksheet listing the following: creativity, abstraction, data, algorithm, networking, beneficial and harmful effects, and impact on society. **Essential knowledge addressed: 7.4.1**C
LO 7.3.1: Analyze the beneficial and harmful effects of computing. [P4] **LO 7.5.1:** Access, manage, and attribute information using effective strategies. [P1] **LO 7.5.2:** Evaluate online and print sources for appropriateness and credibility. [P5]		**Instructional Activity: Copyright and the Law** Students start by generating a list of various legal and ethical components of copyright law. They research articles and write a short paper (with citations) on some aspect of this issue that interests them. **Essential knowledge addressed: 7.3.1** A-D, F, N-Q; **7.5.1** A-C; **7.5.2** A, B
LO 7.1.1: Explain how computing innovations affect communication, interaction, and cognition. [P4]	**Web** "Clive Thompson's New Book Smarter Than You Think| Keen On ..."	**Instructional Activity: Are Computers Making Us Smarter or Dumber?** I pose the question, *Are computers making us smarter or dumber?* Then students watch the video. Next, they search the Internet to find one source that supports each side of the issue, and we hold a class debate. **Essential knowledge addressed: 7.1.1** L-0

Students typically think of computing innovations in terms of how they are personally affected. I open this unit with this formative assessment to help students develop an understanding of global impact.

I carefully add comments to students' worksheets and return them to students to be read.

This is the first of three class debates. The purpose of this debate is to have students see that technology can have different effects depending on the context. Students also get to practice their Internet search skills.

UNIT 7: IMPACT ON SOCIETY

Estimated Time: 8 Hours

BIG IDEA 7 Global Impact

Essential Understandings:
- EU 7.1, EU 7.3, EU 7.4, EU 7.5

Projects and Major Assignments:
- Copyright and the Law Paper
- Class Debates

Guiding Questions

- **How do we define the impact of computing innovations?**
- **What are the beneficial or harmful effects of various computing innovations?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 7.3.1: Analyze the beneficial and harmful effects of computing. [P4] **LO 7.4.1:** Explain the connections between computing and real-world contexts, including economic, social, and cultural contexts. [P1]	**Web** Lanier, "Fixing the Digital Economy"	**Instructional Activity: Is Technology Closing or Widening Economic Inequality?** I pose the question, *Is technology closing or widening economic inequality?* Then students read the article. After reading, we hold a class debate. Finally, students write out their position in one or two paragraphs. **Essential knowledge addressed: 7.31**J; **7.4.1** C, D	*This is the second of three class debates. This activity differs from the ones above and below in that I expect students to write out their thoughts in detail.*
LO 7.1.1: Explain how computing innovations affect communication, interaction, and cognition. [P4] **LO 7.3.1:** Analyze the beneficial and harmful effects of computing. [P4]	**Web** "Danah Boyd at the 2013 NAIS Annual Conference" Turkle, "The Flight from Conversation"	**Instructional Activity: Are Computers Making Our Relationships Stronger or Weaker?** I pose the question, *Are computers making our relationships stronger or weaker?* Then students read the article and watch the video, and we hold a class debate. **Essential knowledge addressed: 7.1.1** A, B; **7.3.1**A	*In this debate, the third in the series, students research a point of view, which will help them in the performance task.*
All of the learning objectives in this unit are addressed.		**Summative Assessment: Short Essay** Students choose one of the debate topics covered in the previous activities and write a one-page essay stating their opinion on the subject. In their essay, they should include how the subject demonstrates the global impact of computing. **All of the unit's essential knowledge statements are addressed.**	*This summative assessment addresses all of the guiding questions for this unit.*

AP Performance Task

ESTIMATED TIME:
8 hours

Explore – Impact of Computing Innovations

At this point, students are ready to complete the Explore Performance Task. I like to save this performance task for last because I feel it is really a capstone for the entire course. Computing innovations use all of the big ideas that students have learned about in the course.

Unit 7 provided a short break from the previous performance task, which I feel helps students focus for this one. In addition, the class has just finished some good debates about the impacts of computing, which will help students make good choices for their Explore innovation.

This task takes my students right up to the deadline for submission; because there is little time to spare, I encourage them at all times to keep backups of their work, as well as draft submissions that are ready to go in case of some external emergency.

Before students begin the performance task, I go through our school's reading protocol (a prescribed way to read a complicated text) with them and make sure that they have no questions. I let students know ahead of time how many class periods we will devote to the performance task to ensure that we meet the number of hours specified in the directions.

I try to keep a cheerful and productive atmosphere in class during this time so that the students can produce their best work.

Resources

All links to online resources were verified before publication. In cases where links are no longer working, we suggest that you try to find the resource by doing a keyword Web search.

General Resources

MacCormick, John. *Nine Algorithms That Changed the Future: The Ingenious Ideas That Drive Today's Computers.* Princeton, NJ: Princeton University Press, 2012.

Unit 1 (Introduction to Computer Science Principles) Resources

"Computation Arts." Concordia University. Video, 3:24. Accessed June 3, 2015. **http://www.concordia.ca/finearts/design/programs/undergraduate/computation-arts-bfa.html**.

"Frogs." NRICH. University of Cambridge. Accessed June 10, 2015. **http://nrich.maths.org/1246**.

"Jacquard Loom Walkthrough." Stacey Harvey Brown. YouTube. Video, 7:13. Accessed June 3, 2015. **https://www.youtube.com/watch?v=f1Zzj9ZBYmQ**.

"Lightbot 2.0." Armor Games. Accessed June 10, 2015. **http://armorgames.com/play/6061/light-bot-20**.

Manjoo, Farhad. "A Bright Side to Facebook's Experiments on Its Users." *New York Times*, July 2, 2014. Accessed June 1, 2015. **http://www.nytimes.com/2014/07/03/technology/personaltech/the-bright-side-of-facebooks-social-experiments-on-users.html**.

McKinley, James C. "Central Park, the Soundtrack." *New York Times*, December 7, 2011. Accessed June 1, 2015. **http://www.nytimes.com/2011/12/08/arts/music/bluebrains-app-central-park-listen-to-the-light.html?pagewanted=all**.

"Pärt Inspired Musical Live Coding Etude Performed in Scratch." YouTube. Video, 2:58. Accessed June 10, 2015. **https://www.youtube.com/watch?v=HyMXCZXjwWg**.

"Player Piano 'The Entertainer.'" YouTube. Video, 3:43. Accessed June 3, 2015. **https://www.youtube.com/watch?v=aseMAEctM1s**.

"Program or Be Programmed." OR Books. Video, 2:19. Accessed June 3, 2015. **http://www.orbooks.com/catalog/program**.

"River Crossing Puzzle." Smart-Kit Puzzles and Games. Accessed June 10, 2015. **http://www.smart-kit.com/s888/river-crossing-puzzle-hard**.

"Scratch Etude - Pärt - SIGCSE Initial Performance Code." Scratch. MIT. Accessed June 10, 2015. **https://scratch.mit.edu/projects/942936**.

Unit 2 (The Internet) Resources

"How the Cyberattack on Spamhaus Unfolded." *New York Times*, March 30, 2013. Accessed June 1, 2015. **http://www.nytimes.com/interactive/2013/03/30/technology/how-the-cyberattack-on-spamhaus-unfolded.html?**.

"How Does the Internet Work?" Data Center Canada. YouTube. Video, 5:37. Accessed June 10, 2015. **https://www.youtube.com/watch?v=i5oe63pOhLI**.

"How Domain Name Servers Work." HowStuffWorks. Accessed June 10, 2015. **http://computer.howstuffworks.com/dns.htm**.

Markoff, John, and Nicole Perlroth. "Attacks Used the Internet Against Itself to Clog Traffic." *New York Times*, March 27, 2013. Accessed June 2, 2015. **http://www.nytimes.com/2013/03/28/technology/attacks-on-spamhaus-used-internet-against-itself.html**.

"Norse – IPViking Live." Accessed June 2, 2015. **http://map.ipviking.com**.

"There and Back Again: A Packet's Tale. How Does the Internet Work?" World Science Festival. Video, 3:29. Accessed June 2, 2015. **http://www.worldsciencefestival.com/2012/06/there_and_back_again_a_packets_tale**.

"United States of Secrets (Part Two)." Frontline. PBS. Accessed June 2, 2015. **http://www.pbs.org/wgbh/pages/frontline/united-states-of-secrets/#united-states-of-secrets-(part-two)**.

Unit 3 (Artificial Intelligence) Resources

"Algorithms Are Taking Over the World: Christopher Steiner at TEDxOrangeCoast." TEDx Talks. YouTube. Video, 11:15. Accessed June 2, 2015. **https://www.youtube.com/watch?v=H_aLU-NOdHM**.

"Chess Exhibit." Computer History Museum. Accessed June 2, 2015. **http://www.computerhistory.org/chess**.

"Course Introduction - Stanford NLP - Professor Dan Jurafsky & Chris Manning." OpenCourseOnline. YouTube. Video, 12:51. Accessed June 2, 2015. **https://www.youtube.com/watch?v=nfoudtpBV68**.

"CS247: Human-Computer Interaction Design Studio." StanfordHCI. YouTube. Video, 5:36. Accessed June 10, 2015. **https://www.youtube.com/watch?v=9Pe9IUAYYTk**.

"Exploring the Epic Chess Match of Our Time." FiveThirtyEight. Accessed June 10, 2015. **http://fivethirtyeight.com/features/the-man-vs-the-machine-fivethirtyeight-films-signals**.

Hoffman, Jan. "Reading Pain in a Human Face." *New York Times*, April 28, 2014. Accessed June 2, 2015. **http://well.blogs.nytimes.com/2014/04/28/reading-pain-in-a-human-face**.

NACLO practice problems. North American Computational Linguistics Olympiad. Accessed June 2, 2015. **http://www.naclo.cs.cmu.edu/practice.html**.

"Rock-Paper-Scissors: You vs. the Computer." *New York Times*. Accessed June 2, 2015. **http://www.nytimes.com/interactive/science/rock-paper-scissors.html**.

"The Watson Trivia Challenge." *New York Times*, June 16, 2010. Accessed June 1, 2015. **http://www.nytimes.com/interactive/2010/06/16/magazine/watson-trivia-game.html**.

"What Is Watson?" IBM. Accessed June 1, 2015. **http://www.ibm.com/smarterplanet/us/en/ibmwatson/what-is-watson.html**.

Unit 4 (Abstraction and Simulation) Resources

"Abstraction and Testing." Beauty and Joy of Computing. University of California, Berkeley. Accessed June 10, 2015. **http://bjc.eecs.berkeley.edu/bjc-r/topic/topic.html?topic=berkeley_bjc%2Fintro_new%2F4-abstraction-testing.topic**.

"Animations." SBEL. University of Wisconsin – Madison. Accessed June 10, 2015. **http://sbel.wisc.edu/Animations**.

"Cisco Binary Game." Accessed June 2, 2015. **http://forums.cisco.com/CertCom/game/binary_game_page.htm**.

Conway's Game of Life. Accessed June 1, 2015. **http://pmav.eu/stuff/javascript-game-of-life-v3.1.1**.

Excerpt from James Glieck's *The Information*. *New York Times*, March 18, 2011. Accessed June 2, 2015. **http://www.nytimes.com/2011/03/20/books/review/excerpt-the-information-by-james-gleick.html?pagewanted=all**.

"Hamming Code (7,4) Introduction to Telecommunications." GRCCtv. YouTube. Video, 11:34. Accessed June 10, 2015. **https://www.youtube.com/watch?v=0FACmnjV7zg**.

Logicly. Accessed June 2, 2015. **http://logic.ly**.

"Online CS Modules: The Central Processing Unit." Virginia Tech. Accessed June 2, 2015. **http://courses.cs.vt.edu/csonline/MachineArchitecture/Lessons/CPU/index.html**.

"Retirement Nest Egg Calculator." Vanguard. Accessed June 2, 2015. **https://retirementplans.vanguard.com/VGApp/pe/pubeducation/calculators/RetirementNestEggCalc.jsf**.

Unit 5 (Data) Resources

"200 Countries, 200 Years, 4 Minutes." Gapminder. Video, 4:47. Accessed June 2, 2015. **http://www.gapminder.org/videos/200-years-that-changed-the-world-bbc**.

"Computer Science Principles Widgets." Baker Franke. Code.org. Accessed June 3, 2015. **https://code.org/educate/csp/aboutTheWidgets**.

"Connected Cars. Connected Homes. Connected Lives." IBM Analytics. YouTube. Video, 4:05. Accessed June 10, 2015. **https://www.youtube.com/watch?v=8R0CPFlU14M**.

"Everything You've Always Wanted to Know About Akinator." Akinator: The Web Genie! Accessed June 3, 2015. **http://en.akinator.com/content/6/everything-you-ve-always-wanted-to-know-about-akinator**.

"Google Flu Trends." Google. Accessed June 3, 2015. **https://www.google.org/flutrends/us/#US**.

"Google Ngram Viewer." Google. Accessed June 3, 2015. **https://books.google.com/ngrams**.

"Locations." ROSALIND. Accessed June 3, 2015. **http://rosalind.info/problems/locations**.

"Online Photo Editor." PIXLR. Accessed June 3, 2015. **http://apps.pixlr.com/editor**.

"THINK." IBM. Accessed June 3, 2015. **http://www.ibm.com/ibm/think/index.html**.

"Thumbnail Gallery." Matplotlib. Accessed June 3, 2015. **http://matplotlib.org/gallery.html**.

Topaz, Maxim. "The Hitchhiker's Guide to Nursing Informatics Theory: Using the Data-Knowledge-Information-Wisdom Framework to Guide Informatics Research." *Online Journal of Nursing Informatics* 17, no. 3 (2013). Accessed June 3, 2015. **http://ojni.org/issues/?p=2852**.

"Wealth and Health of Nations." Gapminder. Accessed June 8, 2015. **https://gapminder.org/world**.

Unit 6 (Intractable Problems) Resources

"158,962,555,217,826,360,000 (Enigma Machine)." Numberphile. YouTube. Video, 11:51. Accessed June 10, 2015. **https://www.youtube.com/watch?v=G2_Q9FoD-oQ**.

"Codes, Ciphers, Encryption and Cryptography." Braingle. Accessed June 2, 2015. **http://www.braingle.com/brainteasers/codes/index.php**.

"Cryptograms." Simon Singh. Accessed June 10, 2015. **http://simonsingh.net/cryptography/cryptograms**.

"Encryption and HUGE Numbers." Numberphile. YouTube. Video, 9:21. Accessed June 2, 2015. **https://www.youtube.com/watch?v=M7kEpw1tn50**.

"Enigma Simulation." Accessed June 2, 2015. **http://enigmaco.de/enigma/enigma.html**.

"Network Solutions SSL Certificate." Video, 2:55. Accessed June 10, 2015. **http://www.qualitywebhostsolutions.com/network-solutions-ssl-certificate/**.

"Simple RSA Key Generation." A Security Site. Accessed June 2, 2015. **http://asecuritysite.com/Encryption/rsa**.

"The Traveling Salesman with Simulated Annealing, R, and Shiny." Todd W. Schneider. Accessed June 2, 2015. **http://toddwschneider.com/posts/traveling-salesman-with-simulated-annealing-r-and-shiny**.

Resources (continued)

Unit 7 (Impact on Society) Resources

“Clive Thompson’s New Book Smarter Than You Think | Keen On ...” TechCrunch. YouTube. Video, 10:51. Accessed June 3, 2015. **https://www.youtube.com/watch?v=5DSm8n7Ir98**.

“Danah Boyd at the 2013 NAIS Annual Conference.” NAIS. YouTube. Video, 20:45. Accessed June 3, 2015. **https://www.youtube.com/watch?v=_YOR69TBxBA**.

Lanier, Jaron. “Fixing the Digital Economy.” *New York Times*, June 8, 2013. Accessed June 3, 2015. **http://www.nytimes.com/2013/06/09/opinion/sunday/fixing-the-digital-economy.html?pagewanted=all**.

Turkle, Sherry. “The Flight from Conversation.” New York Times, April 21, 2012. Accessed June 26, 2015. **http://www.nytimes.com/2012/04/22/opinion/sunday/the-flight-from-conversation.html?pagewanted=all**.