AP® Computer Science Principles

Course Planning and Pacing Guide

Richard Kick

Newbury Park High School ▸ Newbury Park, CA

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

Newbury Park High School ▸ Newbury Park, CA

School

Newbury Park High School is in the Conejo Valley Unified School District, which has 18 elementary schools, six middle schools, and five high schools. The high school is a suburban school approximately 50 miles northwest of Los Angeles, CA. It is commonly known among community members as "Nice People High School."

Student population

There are 2,550 students at Newbury Park High School with the following demographics: 2 percent African American, 8 percent Asian, 21 percent Hispanic, and 68 percent Caucasian. Each year, approximately 95 percent of the school's graduates enter colleges or universities.

Instructional time

The high school offers an intensive block schedule that includes three 95-minute class periods plus two 50-minute class periods, one at the beginning and one at the end of the school day. Classes that meet each school day complete a traditional full-year class in one semester (90 days). Classes that meet every other school day on the alternating block cycle complete a traditional year-long class in one school year (180 days). Computer Science Principles meets every other school day, beginning in August and ending the second week of June.

Student preparation

Students at Newbury Park High School enter Computer Science Principles with a wide variety of backgrounds. Some students have had no formal computer science training. Other students have taken an introduction to HTML and CSS or an introduction to BASIC programming in one of the introductory courses offered at the school. Some students have experienced summer programs that included introductory programming using one of several possible programming languages. We also have a small number of students who have completed AP Computer Science A before entering Computer Science Principles. It is expected that all students entering Computer Science Principles will be knowledgeable about the topics taught in Algebra I, including functions, variables, and expressions.

Instructional Setting (continued)

Primary planning resources

Abelson, Hal, Ken Ledeen, and Harry Lewis. *Blown to Bits: Your Life, Liberty, and Happiness After the Digital Explosion*. New York: Addison-Wesley, 2008. Accessed June 15, 2015. **http://www.bitsbook.com**.

Blown to Bits *provides engaging readings about the digital world and its effects on our society. It is available free as a PDF online and for purchase in print.*

ACM TechNews. Accessed July 14, 2015. **http://technews.acm.org**.

This online collection of articles discusses important research related to computer science.

IntroComputing.org. Accessed July 14, 2015. **http://introcomputing.org**.

This free resource teaches and gives practice in JavaScript programming.

Kick, Richard. "Webpage Programming with JavaScript." Accessed July 22, 2015. **http://kickstyle.net/~rkick/courses/apcs/principles**.

I often use this instructor site to display JavaScript programming examples.

Piazza. Accessed July 14, 20154. **http://piazza.com**.

I use Piazza as my course management system.

"Resources." Mr. Kick. Accessed July 14, 2015. **https://sites.google.com/site/nphsrkick/Home/csprinciples/resources**.

I often use this list of Web resources in my Computer Science Principles course at Newbury Park High School for various activities.

Overview of the Course

Computer Science Principles is designed to encourage a diverse group of students to explore computer science. Rather than limiting this introductory study to just two traditional topics — algorithms and programming — this course introduces students to a broad set of big ideas. These big ideas, which include algorithms and programming, are often summarized using the terms *creativity, abstraction, data, Internet*, and *impact*. In addition, this course emphasizes the use of computational thinking practices for effective learning experiences and problem solving. These practices include connecting, creating, abstracting, analyzing, communicating, and collaborating.

At Newbury Park High School, Computer Science Principles uses an integrated learning approach. Students build their knowledge and understanding through participation in a wide variety of activities and explorations. These experiences are not broken into contiguous blocks of common content. Instead, they are integrated throughout the school year. Before, during, and after explorations, connections are made to the seven big ideas at the core of the course. Activities encourage students to regularly apply the six computational thinking practices to their work.

Although explorations may primarily focus on a single big idea, there is a conscious effort to have students make connections to each of the seven big ideas during each exploration. For example, explorations that are primarily focused on programming are distributed throughout the course as a means to support students in their search for enhanced understanding of all seven big ideas. Exploration themes do not occur at just one time in the course but at various times throughout it. Students revisit themes and build on previous knowledge by exploring the themes at a deeper conceptual level. While working within the introductory theme, students become familiar with terminology, content, and computational tools. Later explorations require students to communicate about their explorations using new terms, content, and tools in addition to the terms, content, and tools previously introduced. By the end of the course, students have been encouraged to seek new resources and share them with their peers to more deeply and rapidly repeat the learning cycle experienced during earlier introductions.

The primary programming language used in this course is JavaScript. However students are exposed to other programming options in the form of block languages and languages that illustrate the connection between blocks and text-based syntax.

As the instructor, I guide students toward personal discoveries and introduce them to computer science topics that are related to current events and their own experiences. I often draw students to these topics through the use of storytelling. The stories shared originate from community experiences, current and former student experiences, and my own experiences. The stories derive from topics related to popular culture, historical events, or any other areas of student interest. Rather than limiting studies to materials I present, I encourage students to pursue personal interests related to presented materials, which often leads to explorations in unanticipated directions. Student-initiated explorations are among the most valuable for both students and the instructor. After each exploration, I organize discussions to facilitate student reflections about what they have learned, what they have yet to learn, and what they wish to further explore in future studies. Both students and the instructor influence the final activities and related requirements associated with final assessments. This method requires me to be familiar with a wide variety of possible destinations to which each exploration can lead.

Explorations for Computer Science Principles are designed to spark interest, curiosity, enthusiasm, and enjoyment. These ambitious goals are usually accomplished through adjustments made because of student contributions to classroom discussions, online posts using the classroom management system, and student journals.

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.
- 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.
- 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.
- 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.
- Justify appropriateness and correctness of a solution, model, or artifact.

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.
- 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 an overall collaborative effort.
- Foster a constructive, collaborative climate by resolving conflicts and facilitating the contributions of a partner or team member.
- Exchange knowledge and feedback with a partner or team member.
- Review and revise their work as needed to create a high-quality artifact.

Pacing Overview

Exploration Theme	Hours of Exploration	Exploration Theme Summary
0: Introduction to CSP	15	Computer science is introduced as a study of the seven big ideas facilitated by the application of the six computational thinking practices. Websites inspire discussions and document creation related to current topics. Students read and discuss articles, respond in journal entries, and create artifacts that summarize their interests and knowledge.
1: Creativity and Computing	25	Students write program code, primarily in JavaScript, throughout the school year. Much of this programming focuses on creating artifacts of interest to individual students. Manipulations range from modifying all pixels in an image to performing algorithmic computations associated with changes in selected collections of pixels. Students write programs to merge images using pixel selection algorithms, create patterns, and combine multiple techniques into a new, higher-level technique. Students analyze and visualize data sets, explore JavaScript application programming interfaces, and create a dynamic website that facilitates the exploration of topics of personal interest.
2: Identifying and Using Abstractions	10	Students are introduced to abstractions used to efficiently create effective programs. Everything that a student says or does involves the use of abstractions. It is important to help students become aware of abstractions and how they affect their ability to understand and navigate the world in which they live. In particular, Computer Science Principles involves abstractions whose application enhances computational capabilities.
3: Using Data	10	Programming is used to help students effectively process and summarize data. Publicly available data sources are used for exploration, the discovery of information, and the creation of new knowledge. Students identify topics that interest them and search the Web for data associated with those topics. Early in the course, students are provided data sources that are stored on the server used for their programming. This method allows students to more easily process data in an effort to discover new information and knowledge. Later in the course, data sources must be entered into the programming environment in an acceptable format. Doing so often requires computational manipulation of the data using text editors, spreadsheets, JavaScript programs, or statistics software packages.

Pacing Overview (continued)

Exploration Theme	Hours of Exploration	Exploration Theme Summary
4: Exploring Algorithms	10	Students present, analyze, and implement algorithms that are designed to accomplish specific tasks related to solving problems of personal interest and motivational artifact creation. While exploring image manipulations, students learn how to create gray scale images, produce enhanced color schemes based on original image color schemes, merge images, and transform images using reflections, rotations, and dilations. Students also generate row, column, diagonal, and checkerboard patterns. They use their knowledge and experience to create new algorithms that accomplish their image manipulation goals. Students also apply their knowledge of algorithms to process data and gain new information and knowledge.
5: Problem Solving With Programming	20	Students use programming to efficiently solve problems. Relatively simple problems that require programming to efficiently obtain a solution are presented to students early in the course. For example, students are asked questions about images in terms of their color properties. Because the images contain thousands of pixels, the students must use computational thinking to answer the questions in a reasonable amount of time. Students are required to write programs that efficiently answer the questions, and they are challenged to pose questions that they and their peers can further explore.
6: Guided Internet Explorations	15	Students explore the characteristics and uses of the Internet. They examine and discuss security and privacy issues related to the Internet. For example, students simulate a binary transmission using paper clips. They create and use Huffman coding trees to encode text messages into binary form and then model the messages using chains or packets of connected paper clips. Students also explore encryption and decryption techniques.
7: Identifying Innovations	20	Students select areas of interest that are affected by computing. They research and report on those areas in written and oral reports and present computational artifacts. Past activities have included a debate about the impact of computing on society, the creation of videos on computing in a variety of fields, and the development of infographics that illustrate computing information that is personally relevant.

EXPLORATION 0: INTRODUCTION TO CSP

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 7 Global Impact

Essential Understandings:
- EU 1.1, EU 1.2, EU 7.1, EU 7.3

Projects and Major Assignments:
- Creating and Using Piazza Accounts
- Using Google Accounts to Create Course Journals
- Reading and Summarizing Articles About Computing

Guiding Questions

- **What is computer science?**
- **How can information be efficiently communicated between small and large groups of people?**
- **How can productive collaboration be effectively facilitated?**
- **How has cloud computing affected our education?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6]	**Web** Piazza	**Instructional Activity: Creating and Using Piazza Accounts** I lead students through the process of creating an account on Piazza and posting introductory information about themselves. They create this self-introduction in the form of a Piazza "Note" and typically include the name by which they prefer to be addressed, their primary interests, and their experiences using a computer as a creative tool. Students also respond to at least two introductory posts by their peers. They can post questions about any course-related content in the form of a Piazza "Question." Students and I can respond and mark it as "Resolved" or "Unresolved" if the originator of the question requires further communication. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A, B, E; **1.2.2** A; **1.2.3** A; **1.2.4** A, B, F	*Students create and use an account on the Piazza course management system using the school-provided Gmail account as identification. This approach allows me to identify users in a consistent manner through the use of a single email address. I give students time in class to use Piazza. Doing so is most important for students who do not have easy access to the Internet outside the classroom.*
		Formative Assessment: Creating and Using Piazza Accounts Every day, students read and respond to posts that their peers and I created as an ongoing formative assessment. Students also reflect on how the use of Piazza affects their ability to gain an education. I lead discussions about perspectives on how computing has affected education over the last three decades. I give students opportunities to explain how computing has affected their personal educational experiences. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A, B, E; **1.2.2** A; **1.2.3** A; **1.2.4** A, B, F	*Piazza provides a tool that summarizes the number of posts created and read by individual students. Although this information can be helpful to the instructor, it does not provide details about the quality of the posts. I must read the posts and determine the extent to which students are meeting course expectations.*

EXPLORATION 0: INTRODUCTION TO CSP

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 7 Global Impact

Essential Understandings:
- EU 1.1, EU 1.2, EU 7.1, EU 7.3

Projects and Major Assignments:
- Creating and Using Piazza Accounts
- Using Google Accounts to Create Course Journals
- Reading and Summarizing Articles About Computing

Guiding Questions

- What is computer science?
- How can information be efficiently communicated between small and large groups of people?
- How can productive collaboration be effectively facilitated?
- How has cloud computing affected our education?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6]	**Web** Google Docs journals	**Instructional Activity: Using Google Accounts to Create Course Journals** Students create a private Google Docs journal that they use throughout the course to share information with me. This journal is shared with only me, not with other students in the class. My available time will determine whether I instruct students to share the journal document as "Can comment" or "Can view." I recommend that the document not be shared using the "Can edit" option, as this setting may lead to confusion as to whether or not I modified the document. In any case, using the "See revision history" file option can clarify when and by whom the document was modified. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A, B, E; **1.2.2** A; **1.2.3** A; **1.2.4** A, B, F

EXPLORATION 0: INTRODUCTION TO CSP

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 7 Global Impact

Essential Understandings:
- EU 1.1, EU 1.2, EU 7.1, EU 7.3

Projects and Major Assignments:
- Creating and Using Piazza Accounts
- Using Google Accounts to Create Course Journals
- Reading and Summarizing Articles About Computing

Guiding Questions

- **What is computer science?**
- **How can information be efficiently communicated between small and large groups of people?**
- **How can productive collaboration be effectively facilitated?**
- **How has cloud computing affected our education?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6]	**Web** Google Docs journals	**Formative Assessment: Using Google Accounts to Create Course Journals** Every week, students write in their journals about their explorations. I use the journals to assess students' writing and monitor their ability to include sufficient detail and depth. Generally, students respond to the following questions: 1. What exploration this week was particularly interesting, was particularly motivational, or inspired a strong sense of personal accomplishment? 2. What exploration this week was particularly challenging, was less interesting, or resulted in significant struggles to accomplish your goals? 3. What activities or ideas did we not explore that you would like to include for future consideration? Additional prompts are included throughout the course to emphasize the focus of each topic explored. Prompts can also include multiple-choice questions. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A, B, E; **1.2.2** A; **1.2.3** A; **1.2.4** A, B, F

Keeping answers secure from peers can be difficult, depending on the configuration of the computer lab. Generally, quiz questions selected for inclusion in the journal document require individual insights and perspectives. This characteristic makes copying the answers of another student impractical.

EXPLORATION 0: INTRODUCTION TO CSP

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 7 Global Impact

Essential Understandings:
- EU 1.1, EU 1.2, EU 7.1, EU 7.3

Projects and Major Assignments:
- Creating and Using Piazza Accounts
- Using Google Accounts to Create Course Journals
- Reading and Summarizing Articles About Computing

Guiding Questions

- **What is computer science?**
- **How can information be efficiently communicated between small and large groups of people?**
- **How can productive collaboration be effectively facilitated?**
- **How has cloud computing affected our education?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
7.1.1: Explain how computing innovations affect communication, interaction, and cognition. [P4] **7.1.2:** Explain how people participate in a problem solving process that scales. [P4] **7.3.1:** Analyze the beneficial and harmful effects of computing. [P4]	**Web** ACM TechNews Google Docs journals	**Instructional Activity: Reading and Summarizing Articles About Computing** I give students time to independently read the titles of a variety of articles highlighted on ACM TechNews. I then ask them to independently select the one article that they find most interesting, read the article, and write a brief summary of it in their journal. I then give students an opportunity to share their insights and discoveries with their table partners and the entire class. As the year progresses, students are expected to improve their ability to use correct computer science terminology and include details associated with computing innovations that are related to the articles. **Essential knowledge addressed: 7.1.1** D, E, M, O; **7.1.2** E; **7.3.1** K
	Print Abelson, Ledeen, and Lewis, chapters 1, 2, and 5, Conclusion, and Appendix **Web** Google Docs journals	**Instructional Activity: Reading and Summarizing *Blown to Bits*** Throughout the course, students read sections of *Blown to Bits* and summarize their findings in small- and large-group discussions and journal posts. I select particular sections of the book to emphasize specific content. Students select other sections based on individual interests. Typically, students read all of chapter 1, although not in a single sitting. Students also read all of chapters 2 and 5, the conclusion, and the appendix. I allow students to read some sections during class time and provide paper copies when necessary to allow for reading outside of class. I give students questions to answer for required readings. For student-selected sections, students summarize the reading in paragraph form and as a bulleted list of major computing concepts learned from the reading. **Essential knowledge addressed: 7.1.1** D, E, M, O; **7.1.2** E; **7.3.1** K

Discussions are characterized as having volunteers "starting the discussion" rather than stating answers. I encourage students to share their perspectives, even if they are not certain they are correct. Comments and replies to statements help all students enhance their perspectives on the topic of focus. Although student-led discussions may address all ideas, no particular essential knowledge statements other than the ones listed here are required during this exploration.

I usually facilitate student discussions by grouping students in table groups. Large-group discussions of what each group summarizes take place among the entire class. Students create written summaries using Google Docs that are shared with the group. Students transfer the information to Piazza as a post to allow the summaries to remain persistent and accessible by all students throughout the remainder of the course.

EXPLORATION 0: INTRODUCTION TO CSP

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 7 Global Impact

Essential Understandings:
- EU 1.1, EU 1.2, EU 7.1, EU 7.3

Projects and Major Assignments:
- Creating and Using Piazza Accounts
- Using Google Accounts to Create Course Journals
- Reading and Summarizing Articles About Computing

Guiding Questions

- What is computer science?
- How can information be efficiently communicated between small and large groups of people?
- How can productive collaboration be effectively facilitated?
- How has cloud computing affected our education?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from this exploration are addressed.	**Web** Google Docs journals Google Forms Piazza	**Summative Assessment: Introduction to CSP** Students answer multiple-choice and free-response questions about particular topics presented and discussed in class. Of course, because the content is presented and discussed throughout the course, portions of the assessment are integrated into several assessments given on a monthly basis. This method helps students keep all concepts current as they prepare for the end-of-year assessment. Students submit their summative assessments in the form of Piazza posts that are private and can be seen only by the instructor, as submissions to Google Forms, or as additions to their student journal. **All of the exploration's essential knowledge statements are addressed.**

This summative assessment addresses all the guiding questions for this exploration.

Teachers more comfortable with a traditional approach to organizing course materials could organize each exploration and its corresponding summative assessment into a single unbroken block of time. However, this approach reduces opportunities for students to make connections and be creative.

EXPLORATION 1: CREATIVITY AND COMPUTING

Estimated Time: 25 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 1.1, EU 1.2, EU 1.3, EU 2.1, EU 2.2, EU 4.1, EU 5.1, EU 5.5, EU 7.2

Projects and Major Assignments:
▶ Introduction to Programming ▶ The Chaos Game ▶ Using Online Tools to Generate Art ▶ All-School Art Show

Guiding Questions

▶ What are digital images, and how can they be created and manipulated? ▶ How can mathematics be used to generate images that are artistically appealing? ▶ How can computational artifacts be used to affect a community? ▶ What is recursion, and how can its use enhance the creation of computational artifacts?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **LO 1.3.1:** Use computing tools and techniques for creative expression. [P2] **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** IntroComputing .org Piazza "Rihanna — Diamonds (Live on SNL)"	**Instructional Activity: Introduction to Programming** Students work in pairs as "drivers" and "navigators" to complete the "Introduction and Code" and "Digital Images" exercises on IntroComputing.org, along with extension exercises posted on Piazza. The *traditional for loop* is introduced and applied to the creation of pixel patterns. The *modulus operator %* is defined and used to make patterns of pixels in row, column, and checkerboard patterns. Students use solid colors and pixels obtained from images to create the patterns. They use binary representations of data and binary operators to modify pixel colors. Students explore base systems and conversions between bases. We regularly discuss the manipulation of images as a class, in small groups, and on Piazza. Students post screenshots of the output produced and the code they used to generate it. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-E; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** A, C; **2.1.1** A-G; **2.1.2** D; **2.2.1** B, C; **2.2.2** A, B; **2.2.3** A, K; **5.1.1** A-F; **5.5.1** A-G; **7.2.1** G

Encourage students to explore the use of the binary operators &, |, and ^ before they know their definitions and properties. After student explorations, formally define the operators and their properties. Students should make connections between what they observed in their images and the properties associated with the operators. Examples of digital image manipulations in popular culture, such as the background images in the Rihanna performance, can increase motivation and spark discussion.

EXPLORATION 1: CREATIVITY AND COMPUTING

Estimated Time: 25 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 1.1, EU 1.2, EU 1.3, EU 2.1, EU 2.2, EU 4.1, EU 5.1, EU 5.5, EU 7.2

Projects and Major Assignments:

▶ Introduction to Programming ▶ The Chaos Game ▶ Using Online Tools to Generate Art ▶ All-School Art Show

Guiding Questions

▶ **What are digital images, and how can they be created and manipulated?** ▶ **How can mathematics be used to generate images that are artistically appealing?** ▶ **How can computational artifacts be used to affect a community?** ▶ **What is recursion, and how can its use enhance the creation of computational artifacts?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
(continued from previous page) **LO 2.2.1:** Develop an abstraction when writing a program or creating other computational artifacts. [P2] **LO 2.2.2:** Use multiple levels of abstraction to write programs. [P3] **LO 2.2.3:** Identify multiple levels of abstractions that are used when writing programs. [P3] **LO 5.1.1:** Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. [P2] **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** IntroComputing .org Piazza	**Formative Assessment: Introduction to Programming** Students write JavaScript code within IntroComputing.org text areas to generate images. They post their images and corresponding code on Piazza. Students then comment on and ask questions about their classmates' posts. I give students credit for their posts if they communicate information that helps the reader to better understand the process used to develop an image. I also give them credit if they ask specific questions that help themselves and their peers better understand the content of existing posts. Students create some images by writing code collaboratively in pairs. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-E; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** A, C; **2.1.1** A-G; **2.1.2** D; **2.2.1** B, C; **2.2.2** A, B; **2.2.3** A, K; **5.1.1** A-F; **5.5.1** A-G; **7.2.1** G

I usually assign the images that are the most algorithmically challenging to produce as pair programming assignments. I assign images that encourage students to express themselves creatively as independent projects.

EXPLORATION 1: CREATIVITY AND COMPUTING

Estimated Time: 25 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 1.1, EU 1.2, EU 1.3, EU 2.1, EU 2.2, EU 4.1, EU 5.1, EU 5.5, EU 7.2

Projects and Major Assignments:

▸ Introduction to Programming ▸ The Chaos Game ▸ Using Online Tools to Generate Art ▸ All-School Art Show

Guiding Questions

▸ **What are digital images, and how can they be created and manipulated?** ▸ **How can mathematics be used to generate images that are artistically appealing?** ▸ **How can computational artifacts be used to affect a community?** ▸ **What is recursion, and how can its use enhance the creation of computational artifacts?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4]	**Web** Paul, Jody, Rich Kick, and Paul Tymann, "AP CSP Curriculum Module — Creativity and Global Impact" IntroComputing .org	**Instructional Activity: The Chaos Game** Students start with code from the Creativity and Global Impact Curriculum Module found under the resources section on the AP Teacher Community for Computer Science Principles. Students should independently research the chaos game using a search engine and then modify and add to the code provided to create original extensions of the chaos game that are personally relevant. Most recently, students have used pixel properties from images rather than solid colors to plot chaos figures. They accomplished this by writing code using the IntroComputing.org website's "Custom Bluescreen" text area to load images that reside on their computer. The results were fractal images combining properties of multiple existing images that were personally meaningful to the students. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-E; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D

EXPLORATION 1: CREATIVITY AND COMPUTING

Estimated Time: 25 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 1.1, EU 1.2, EU 1.3, EU 2.1, EU 2.2, EU 4.1, EU 5.1, EU 5.5, EU 7.2

Projects and Major Assignments:

▶ Introduction to Programming ▶ The Chaos Game ▶ Using Online Tools to Generate Art ▶ All-School Art Show

Guiding Questions

▶ **What are digital images, and how can they be created and manipulated?** ▶ **How can mathematics be used to generate images that are artistically appealing?** ▶ **How can computational artifacts be used to affect a community?** ▶ **What is recursion, and how can its use enhance the creation of computational artifacts?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4]	**Web** "AP CSP Curriculum Module — Creativity and Global Impact" Piazza	**Formative Assessment: Implementing and Extending the Chaos Game** Students write code to implement a variety of versions of the chaos game. Multiple examples are given in the Creativity Curriculum Module, but students should create images different than the examples. Differences may include variations such as color, vertex number and location, move to ratios, probabilities associated with the frequency of selecting particular vertices, and plotted pixel colors using images selected by students. Students should post at least three images to Piazza: one that was the result of collaborative work with a partner, one that was the result of individual work, and one that illustrates insights gained and progress made during their individual explorations. Students post their images and corresponding code on Piazza, and they comment and ask questions about their peers' posts. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-E; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D

As I review students' code, I provide comments and guidance as needed to ensure that the students are progressing in their understanding of programming constructs.

EXPLORATION 1: CREATIVITY AND COMPUTING

Estimated Time: 25 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 1.1, EU 1.2, EU 1.3, EU 2.1, EU 2.2, EU 4.1, EU 5.1, EU 5.5, EU 7.2

Projects and Major Assignments:

▶ Introduction to Programming ▶ The Chaos Game ▶ Using Online Tools to Generate Art ▶ All-School Art Show

Guiding Questions

▶ **What are digital images, and how can they be created and manipulated?** ▶ **How can mathematics be used to generate images that are artistically appealing?** ▶ **How can computational artifacts be used to affect a community?** ▶ **What is recursion, and how can its use enhance the creation of computational artifacts?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **LO 4.1.1:** Develop an algorithm for implementation in a program. [P2] **LO 4.1.2:** Express an algorithm in a language. [P5]	**Web** "Blockly Games: Turtle" Mr. doob Pencil Code Recursive Drawing	**Instructional Activity: Using Online Tools to Generate Art** In this activity, I introduce students to the programming sites listed in the Materials section of this activity. I encourage students to explore each site and use them to generate unique artifacts. I introduce recursion using graphic examples created with Recursive Drawing, plus nongraphic examples of recursion in the form of mathematical definitions, such as exponentiation. Students use a shape they have created to generate recursive images by dragging a copy of the original shape into the drawing region it occupies. The software replicates the copy and the modifications associated with it. Shape modifications can include changes in size and rotational orientation. In pairs, students produce complex, self-similar images using relatively simplistic algorithms and describe the steps they used to generate the recursively generated graphics. **Essential knowledge addressed:** **1.1.1** A, B; **1.2.1** A-E; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **4.1.1** A, B, E, H, I; **4.1.2** A-I

EXPLORATION 1: CREATIVITY AND COMPUTING

Estimated Time: 25 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 1.1, EU 1.2, EU 1.3, EU 2.1, EU 2.2, EU 4.1, EU 5.1, EU 5.5, EU 7.2

Projects and Major Assignments:

▶ Introduction to Programming ▶ The Chaos Game ▶ Using Online Tools to Generate Art ▶ All-School Art Show

Guiding Questions

▶ What are digital images, and how can they be created and manipulated? ▶ How can mathematics be used to generate images that are artistically appealing? ▶ How can computational artifacts be used to affect a community? ▶ What is recursion, and how can its use enhance the creation of computational artifacts?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **LO 4.1.1:** Develop an algorithm for implementation in a program. [P2] **LO 4.1.2:** Express an algorithm in a language. [P5]	**Web** "Blockly Games: Turtle" Mr. doob Pencil Code Recursive Drawing	**Formative Assessment: Using Online Tools to Generate Art** Students independently write code to generate images possessing the characteristics described in prompts that I provide to them. Examples of characteristics requested are horizontal, vertical, and rotational symmetries like those characteristics exhibited in typical functions studied in Algebra I, as well as self-similarity like the characteristics possessed by many recursively generated graphics. Students post their images and corresponding code on Piazza. They must identify the tools and techniques used to generate the graphics, and they also must comment on and ask questions about their peers' posts. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-E; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **4.1.1** A, B, E, H, I; **4.1.2** A-I

As I review students' code, I provide comments and guidance as needed to ensure that the students are progressing in their understanding of programming constructs.

EXPLORATION 1: CREATIVITY AND COMPUTING

Estimated Time: 25 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 1.1, EU 1.2, EU 1.3, EU 2.1, EU 2.2, EU 4.1, EU 5.1, EU 5.5, EU 7.2

Projects and Major Assignments:

- Introduction to Programming ▸ The Chaos Game ▸ Using Online Tools to Generate Art ▸ All-School Art Show

Guiding Questions

▸ What are digital images, and how can they be created and manipulated? ▸ How can mathematics be used to generate images that are artistically appealing? ▸ How can computational artifacts be used to affect a community? ▸ What is recursion, and how can its use enhance the creation of computational artifacts?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.3.1: Use computing tools and techniques for creative expression. [P2]		**Instructional Activity: Exploring Creativity Associated with Digital Artifacts** Students divide themselves into collaborative groups based on personal interests in the following types of digital artifacts: images, audio, video, and animations. We use approximately three hours of class time to research, using the World Wide Web, how the chosen type of artifact has transformed industries. They must find some technical details about how the artifact type is created and how it is used to transform the industry that they identify. They also should find details about how data are associated with the creation and/or use of the artifact. I let students do research during class to accommodate those who do not have Internet access at home. **Essential knowledge addressed: 1.3.1** A-E
		Formative Assessment: Exploring Creativity Associated with Digital Artifacts Student groups present their discoveries to the class using artifacts, styles, and techniques of their choosing. Each group has the flexibility to present in a way that best aligns with its members' talents and interests. Some groups create videos using video recording and editing tools, some create songs with digital recording and editing software, and others use graphics software. In the past, presentations have included skits, song performances, videos, and program demonstrations. I encourage students to be creative and build on personal interests and experiences. I assess presentations based on the extent (high, medium, or low) to which they address each of the requirements listed in the preceding instructional activity. I also offer potential bonus points for creativity and originality. **Essential knowledge addressed: 1.3.1** A-E

After each presentation I encourage students to ask questions about and share personal experiences with the types of artifacts that were presented.

EXPLORATION 1: CREATIVITY AND COMPUTING

Estimated Time: 25 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 1.1, EU 1.2, EU 1.3, EU 2.1, EU 2.2, EU 4.1, EU 5.1, EU 5.5, EU 7.2

Projects and Major Assignments:
▶ Introduction to Programming ▶ The Chaos Game ▶ Using Online Tools to Generate Art ▶ All-School Art Show

Guiding Questions

▶ **What are digital images, and how can they be created and manipulated?** ▶ **How can mathematics be used to generate images that are artistically appealing?** ▶ **How can computational artifacts be used to affect a community?** ▶ **What is recursion, and how can its use enhance the creation of computational artifacts?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 5.1.1: Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. [P2] **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** IntroComputing .org	**Instructional Activity: All-School Art Show** Students work independently to analyze program code that generates digital fractal images on IntroComputing.org and identify portions of the code responsible for specific characteristics of the images, such as color, size, shape, and complexity. They create and implement algorithms designed to produce new images having the characteristics that they desire. Their goal is to create artifacts that are unique and appealing to themselves and to their peers. The process of creating fractal images includes generating color schemes and shapes using mathematical functions that determine the properties of the images. Each student creates a digital work of art through programming and submits the work as an entry in the all-school art show. Students' art is judged based on originality, complexity, and visual appeal. **Essential knowledge addressed: 5.1.1** A-F; **5.5.1** A-G; **7.2.1** G
All of the learning objectives from this exploration are addressed.	**Web** Piazza	**Summative Assessment: Creativity and Computing** I give students instructions outlining the characteristics of an image they are to create. The image creation process begins with an image that I provide. Students write code to generate image modifications in the form of regular pixel patterns, color modifications, and fractals. To submit their final image, students must take a screenshot of the image and submit both the screenshot and the code used to generate the final image. Students submit their summative assessments in the form of Piazza posts that are private and can only be seen by the instructor or as additions to their student journal. **All of the exploration's essential knowledge statements are addressed.**

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

EXPLORATION 2: IDENTIFYING AND USING ABSTRACTIONS

Estimated Time: 10 Hours

BIG IDEA 2 Abstraction
BIG IDEA 5 Programming

Essential Understandings:
- EU 2.2, EU 5.1, EU 5.3, EU 5.4

Projects and Major Assignments:
- Identifying and Describing Abstractions in Daily Life
- Identifying and Describing Abstractions in Programming
- Using Abstractions to Write Programs

Guiding Questions
- **What is an abstraction?**
- **How does the use of abstractions affect the program development process?**
- **What abstractions are provided in professional application programming interfaces (APIs)?**

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]	**Web** "The Art of Abstraction — Computerphile"	**Instructional Activity: Identifying and Describing Abstractions in Daily Life** I ask students to define the term *cookie* and to work in pairs to agree on a definition. I randomly select students to share their definitions. I then provide examples of cookies that do not match their definitions and emphasize how difficult it is to fully describe a cookie. Independently, students select an everyday term (*car, sandwich, book*) and write a detailed definition that they present to a partner. After each presentation, the partner asks probing questions about the definition, requiring the student to add more detail. Eventually, students realize the concept is much more complex than originally thought. Students switch roles and repeat. I then guide a whole-class discussion summarizing the need for abstractions in order to reduce complexity and facilitate communication about ideas. **Essential knowledge addressed: 2.2.1** A, B	*The concept of a cookie (a type of dessert) is used to illustrate a relatively simple idea that is incredibly difficult to precisely define without excluding the types of cookies that one would want included in the definition. Some students may define cookie in terms of a computer science concept. Acknowledge the interpretation as valid and then redirect students to a conversation about dessert. Students can watch the YouTube video listed under Materials to solidify their understanding.*
	Web Google Docs journals	**Formative Assessment: Identifying and Describing Abstractions in Daily Life** Students independently write a summary about the preceding activity and post it to their Google Docs journal. They describe their choice of concept, their initial definition, the difficulties they and their partner encountered when exploring exceptions to the definition, and the added complexity needed to accurately represent the concept. I then ask students to summarize the importance of abstractions for facilitating communication. Students also should make connections to communications associated with programming. **Essential knowledge addressed: 2.2.1** A, B	*As I review students' summaries, I provide comments and guidance as needed to ensure that the students are progressing in their understanding of abstractions.*

EXPLORATION 2: IDENTIFYING AND USING ABSTRACTIONS

Estimated Time: 10 Hours

BIG IDEA 2 Abstraction
BIG IDEA 5 Programming

Essential Understandings:

- EU 2.2, EU 5.1, EU 5.3, EU 5.4

Projects and Major Assignments:

- Identifying and Describing Abstractions in Daily Life
- Identifying and Describing Abstractions in Programming
- Using Abstractions to Write Programs

Guiding Questions

- **What is an abstraction?**
- **How does the use of abstractions affect the program development process?**
- **What abstractions are provided in professional application programming interfaces (APIs)?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 2.2.2: Use multiple levels of abstraction to write programs. [P3] **LO 2.2.3:** Identify multiple levels of abstractions being used when writing programs. [P3]	**Web** "Google Maps JavaScript API" Piazza "YouTube Data API: JavaScript Code Samples"	**Instructional Activity: Identifying and Describing Abstractions in Programming** I pair students and give them program code that uses the Google Maps API (application programming interface) and YouTube Data API. Students read and identify the abstractions used in the code in the form of a Piazza post. They read appropriate API documentation to fully understand the how the abstractions are represented and how they should be properly applied when programming. This activity helps to prepare students for similar writing associated with the Create Performance Task. **Essential knowledge addressed: 2.2.2** A, B; **2.2.3** A-D
	Web "Google Maps JavaScript API" Piazza "YouTube Data API: JavaScript Code Samples"	**Formative Assessment: Identifying and Describing Abstractions in Programming** Students individually write a summary of their choice of an abstraction used in a program that I provide. The abstraction should correspond to code that is used to provide some significant feature in the program provided. The summary must include the purpose of the abstraction, any input and output associated with the use of the abstraction, and any algorithms used by the abstraction to accomplish its related task. The summary must be written as a Piazza post that is private and can only be seen by the instructor. **Essential knowledge addressed: 2.2.2** A, B; **2.2.3** A-D

As I review abstraction summaries, I provide comments in the form of private Piazza posts that guide students toward improvement in their ability to describe abstractions in detail.

EXPLORATION 2: IDENTIFYING AND USING ABSTRACTIONS

Estimated Time: 10 Hours

BIG IDEA 2 Abstraction
BIG IDEA 5 Programming

Essential Understandings:
- EU 2.2, EU 5.1, EU 5.3, EU 5.4

Projects and Major Assignments:
- Identifying and Describing Abstractions in Daily Life
- Identifying and Describing Abstractions in Programming
- Using Abstractions to Write Programs

Guiding Questions

- **What is an abstraction?**
- **How does the use of abstractions affect the program development process?**
- **What abstractions are provided in professional application programming interfaces (APIs)?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 5.1.1: Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. [P2] **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] **LO 5.4.1:** Evaluate the correctness of a program. [P4]	**Web** "Google Maps JavaScript API" JSFiddle	**Instructional Activity: Using Abstractions to Write Programs** I give students an API (such as the Google Maps JavaScript API) and provide examples of how to read it. I demonstrate how a program feature (e.g., placing a marker on a map at a specific location) is implemented by searching the API for programming abstractions that facilitate the implementation of the feature. Students read the related information about the abstraction, use the sample code provided in the API illustrating the use of the abstraction, and then add to and modify the sample code in a demo program. Modifying the sample code provides the desired result in the sample program (displays a marker in the correct location). Students write program code in collaborative pairs using JSFiddle for creative expression and problem solving. **Essential knowledge addressed: 5.1.1** A-F; **5.1.2** A-J; **5.1.3** A-F; **5.3.1** A-O; **5.4.1** A-E
	Web "Google Maps JavaScript API" JSFiddle Piazza	**Instructional Activity: Using Abstractions to Write Programs** Students individually write programs in JSFiddle that interest them using the Google Maps JavaScript API. Students are encouraged to search the Web for, and to use, additional APIs that they find interesting. They select features that can be implemented using API functionality and write the code that accomplishes the desired results. Students comment their code to provide the reader with insight into the abstractions that are used and how they help the programmer accomplish goals. Students share their work with peers by posting their JSFiddle URLs on Piazza. **Essential knowledge addressed: 5.1.1** A-F; **5.1.2** A-J; **5.1.3** A-F; **5.3.1** A-O; **5.4.1** A-E

Commenting code appropriately is emphasized during this activity. Writing detailed descriptions of the abstractions used helps prepare students for the Create — Applications from Ideas Performance Task.

EXPLORATION 2: IDENTIFYING AND USING ABSTRACTIONS

Estimated Time: 10 Hours

BIG IDEA 2 Abstraction
BIG IDEA 5 Programming

Essential Understandings:

- EU 2.2, EU 5.1, EU 5.3, EU 5.4

Projects and Major Assignments:

- Identifying and Describing Abstractions in Daily Life
- Identifying and Describing Abstractions in Programming
- Using Abstractions to Write Programs

Guiding Questions

- **What is an abstraction?**
- **How does the use of abstractions affect the program development process?**
- **What abstractions are provided in professional application programming interfaces (APIs)?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from this exploration are addressed.	**Web** Google Docs journals Piazza	**Summative Assessment: Identifying and Using Abstractions for Enhanced Creativity and Problem Solving** I give students program code and a general description of the purpose of the program as a Piazza post. They must identify an abstraction used in the program, as well as create and use an abstraction that expands the problem-solving capabilities of the program. Students must provide a written summary of how the identified abstraction and the student-created abstraction are used to accomplish the program's purpose. They share their code, running program screen capture, and abstraction summaries with the instructor in the form of an addition to their Google Docs journal. **All of the exploration's essential knowledge statements are addressed.**

This summative assessment addresses all the guiding questions for this exploration.

It is important to remind students that abstractions are simplified representations of concepts. APIs provide abstractions in the form of procedures designed to accomplish tasks related to the abstraction. Students are encouraged to divide their desired functionality into logical blocks and to implement that functionality as procedures.

EXPLORATION 3: USING DATA

Estimated Time: 10 Hours

BIG IDEA 3 Data and Information
BIG IDEA 5 Programming

Essential Understandings:
- EU 3.1, EU 3.2, EU 3.3, EU 5.5

Projects and Major Assignments:
- Finding Appropriate Online Data Sources ▶ Collecting, Cleaning, and Formatting Data ▶ Visualizing Data ▶ "You Can't Say That on the Internet"

Guiding Questions

▶ Which data sets that are associated with topics of personal interest are freely available online? ▶ What is the difference between data, information, and knowledge? ▶ How can data be cleaned and formatted to facilitate the effective extraction of information?

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 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, notation, and precise language. [P5] **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 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]	**Web** Mr. Kick, "Resources"	**Instructional Activity: Finding Appropriate Online Data Sources** I introduce a variety of online data sources using the "Resources" page on my course website. Students actively search the Web in pairs (selected based on common interests) to find URLs for data sources that are relevant to topics of their choice. Students collect data from their chosen online sources using techniques that are appropriate for the data format that is available. Note that students may find several databases that are very difficult to use. They also may find it difficult to find freely available data sources associated with financial resources. Resources that I provide represent a compromise between individual interests and usable data sources. **Essential knowledge addressed: 3.1.1** A-E; **3.1.2** A-F; **3.1.3** A-E; **3.2.1** A-I; **3.2.2** A-H; **5.5.1** H-J

EXPLORATION 3: USING DATA

Estimated Time: 10 Hours

BIG IDEA 3 Data and Information
BIG IDEA 5 Programming

Essential Understandings:
- EU 3.1, EU 3.2, EU 3.3, EU 5.5

Projects and Major Assignments:
- Finding Appropriate Online Data Sources ▸ Collecting, Cleaning, and Formatting Data ▸ Visualizing Data ▸ "You Can't Say That on the Internet"

Guiding Questions

▸ Which data sets that are associated with topics of personal interest are freely available online? ▸ What is the difference between data, information, and knowledge? ▸ How can data be cleaned and formatted to facilitate the effective extraction of information?

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 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, notation, and precise language. [P5] **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 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]	**Web** Piazza	**Formative Assessment: Finding Appropriate Online Data Sources** Students work in pairs to summarize, using a Piazza post, their topic of interest and their processes for obtaining a data source or data sources. They also summarize the content of the data and potential techniques they may need to use in order to obtain relevant information and knowledge from the data. **Essential knowledge addressed: 3.1.1** A-E; **3.1.2** A-F; **3.1.3** A-E; **3.2.1** A-I; **3.2.2** A-H; **5.5.1** H-J	*As I review students' summaries, I provide comments and guidance as needed to ensure that the students are progressing in their understanding of obtaining appropriate data sources.*
	Local Software Fathom Dynamic Data Software **Web** Mr. Kick, "Resources"	**Instructional Activity: Collecting, Cleaning, and Formatting Data** I demonstrate the process of cleaning and formatting data in order to obtain new insights and knowledge. This cleaning often involves the removal of portions of the data that is not relevant to the area of interest. I use a variety of tools, such as a text editor, a spreadsheet, a professional statistics software package called Fathom, and JavaScript programs that I have written to clean and format the data. Students, in the same pairs used in the previous instructional activity, discuss the techniques they will need to clean and format their data. They divide up the tasks to more efficiently prepare the data for information extraction. They apply the techniques discussed in class to their own data sets. **Essential knowledge addressed: 3.1.1** A-E; **3.1.2** A-F; **3.1.3** A-E; **3.2.1** A-I; **3.2.2** A-H; **5.5.1** H-J	*Cleaning and formatting data uses techniques that vary based on the nature of the data used. The file type used to store the data will also affect which techniques are used to manipulate the data.*

EXPLORATION 3: USING DATA

Estimated Time: 10 Hours

BIG IDEA 3 Data and Information
BIG IDEA 5 Programming

Essential Understandings:
- EU 3.1, EU 3.2, EU 3.3, EU 5.5

Projects and Major Assignments:
- Finding Appropriate Online Data Sources ▸ Collecting, Cleaning, and Formatting Data ▸ Visualizing Data ▸ "You Can't Say That on the Internet"

Guiding Questions

▸ **Which data sets that are associated with topics of personal interest are freely available online?** ▸ **What is the difference between data, information, and knowledge?** ▸ **How can data be cleaned and formatted to facilitate the effective extraction of information?**

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 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, notation, and precise language. [P5] **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 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]	**Web** Piazza	**Formative Assessment: Collecting, Cleaning, and Formatting Data** Each student pair will summarize, using a Piazza post, how they obtained, cleaned, and formatted the data they selected. They must describe in detail the processes they used so that someone unfamiliar with the data could replicate their work. They must also describe any difficulties they encountered while processing the data, as well as how they collaborated to resolve those difficulties. **Essential knowledge addressed: 3.1.1** A-E; **3.1.2** A-F; **3.1.3** A-E; **3.2.1** A-I; **3.2.2** A-H; **5.5.1** H-J	*As I review students' summaries, I provide comments and guidance as needed to ensure that the students are progressing in their understanding of appropriate techniques for obtaining, cleaning, and formatting data.*
	Local Software Fathom Dynamic Data Software Spreadsheet software **Web** IntroComputing .org	**Instructional Activity: Visualizing Data** I demonstrate how to search clean, formatted data for patterns and information. I use a spreadsheet program, Fathom Dynamic Data Software, and instructor-written JavaScript programs to create appropriate visualizations and numerical summaries of the data. Students use data manipulation exercises on IntroComputing.org that explore the Social Security Administration's names data from 1880 to 2014. These data summaries reveal new information and knowledge that can be extracted from data. I also share statistical insights to help students make appropriate choices for their own visualizations and numerical summaries. Students use the tools of their choice to process their data and gain insights that were not available through simply observing the data in its original form. **Essential knowledge addressed: 3.1.1** A-E; **3.1.2** A-F; **3.1.3** A-E; **3.2.1** A-I; **3.2.2** A-H; **5.5.1** H-J	

EXPLORATION 3: USING DATA

Estimated Time: 10 Hours

BIG IDEA 3 Data and Information
BIG IDEA 5 Programming

Essential Understandings:

- EU 3.1, EU 3.2, EU 3.3, EU 5.5

Projects and Major Assignments:

- Finding Appropriate Online Data Sources ▶ Collecting, Cleaning, and Formatting Data ▶ Visualizing Data ▶ "You Can't Say That on the Internet"

Guiding Questions

▶ **Which data sets that are associated with topics of personal interest are freely available online?** ▶ **What is the difference between data, information, and knowledge?** ▶ **How can data be cleaned and formatted to facilitate the effective extraction of information?**

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 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, notation, and precise language. [P5] **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 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]	**Local Software** Fathom Dynamic Data Software Spreadsheet software **Web** Google Drive IntroComputing.org	**Formative Assessment: Visualizing Data** Student pairs create and submit an artifact in the form of a PDF file that visually and numerically represents information and knowledge gained from analyzing the data they selected. Students may use data from IntroComputing.org if they wish. This artifact must consist of a combination of artifacts generated during the students' data-processing activities. Students submit the artifact by uploading the PDF to Google Drive and sharing it with my Google school account. **Essential knowledge addressed: 3.1.1** A-E; **3.1.2** A-F; **3.1.3** A-E; **3.2.1** A-I; **3.2.2** A-H; **5.5.1** H-J

It is important to emphasize that the visualizations created should help the reader to better understand information associated with the data. I use this opportunity to emphasize to students how communications are enhanced through the use of the Internet and the World Wide Web. Student submissions are completely paperless.

EXPLORATION 3: USING DATA

Estimated Time: 10 Hours

BIG IDEA 3 Data and Information
BIG IDEA 5 Programming

Essential Understandings:

- EU 3.1, EU 3.2, EU 3.3, EU 5.5

Projects and Major Assignments:

- Finding Appropriate Online Data Sources ▸ Collecting, Cleaning, and Formatting Data ▸ Visualizing Data ▸ "You Can't Say That on the Internet"

Guiding Questions

▸ **Which data sets that are associated with topics of personal interest are freely available online?** ▸ **What is the difference between data, information, and knowledge?** ▸ **How can data be cleaned and formatted to facilitate the effective extraction of information?**

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** Abelson, Ledeen, and Lewis, chapters 3 and 7 **Web** Google Docs	**Instructional Activity: "You Can't Say That on the Internet"** I give students class time to read portions of Chapters 3 and 7 of *Blown to Bits*. I introduce a question about the reading and give table partners 30 seconds each to answer while their partner listens. After each partner presents an answer, the partners have another 60 seconds to agree on and record their answer in a Google Doc. This structured learning technique is repeated for each question related to the reading. Time is adjusted depending on the complexity of the question presented. After all questions have been answered, table groups share their Google Docs and discuss similarities, differences, and suggestions for a common group answer to each question. Groups share their answers in a class discussion and modify their answers as needed. **Essential knowledge addressed: 3.3.1** A-I
All of the learning objectives from this exploration are addressed.	**Local Software** Fathom Dynamic Data Software Spreadsheet software **Web** Google Docs Google Forms	**Summative Assessment: Using Data** I give students a large data set that I distribute as a shared Google Doc. They must independently clean, filter, format, and visualize the data to answer specific questions about the data and the processes they used to manipulate the data. The questions are provided in a Google Form, which students use to submit their answers. **All of the exploration's essential knowledge statements are addressed.**

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

This assessment can significantly vary in difficulty based on the size and format of the data provided to the students. The validity of the assessment can be improved by sharing different versions of the data set that lead to different correct answers for each set.

EXPLORATION 4: EXPLORING ALGORITHMS

Estimated Time: 10 Hours

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms

Essential Understandings:
- EU 1.1, EU 1.2, EU 4.1, EU 4.2

Projects and Major Assignments:
- Creating Algorithms to Manipulate Images
- Huffman Coding Trees
- Comparing Algorithms for Image File Creation
- Designing Algorithms for Map Creation

Guiding Questions

- **What is an algorithm?**
- **What are common algorithms used for digital image manipulation?**
- **How can algorithms be evaluated for correctness and efficiency?**
- **How can messages be translated between multiple representations?**
- **How is map location data used to obtain practical information?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **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 4.2.1:** Explain the difference between algorithms that run in a reasonable time and those that do not run in a reasonable time. [P1]	**Web** "Algorithms" "How Search Works: From Algorithms to Answers" IntroComputing.org	**Instructional Activity: Creating Algorithms to Manipulate Images** I introduce the concept of algorithms by having a volunteer instruct a peer on how to leave the classroom. The peer must respond to relatively low-level commands such as "stand," "turn," and "walk" but to ignore high-level commands such as "go to the door." Students generate an algorithm designed to guide their peer out of the room. I present the Google Inside Search "How Search Works" site as an example of how algorithms are used to accomplish powerful computing tasks. Next, students choose images from IntroComputing.org to manipulate. They create and implement algorithms for common image manipulations such as gray scale, reflections, and rotations. Each algorithm is analyzed during small-group and whole-class discussions. Students also create new algorithms that manipulate pixel patterns in unfamiliar ways. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A, B, E; **1.2.2** A; **1.2.3** A; **1.2.4** A, B, F; **1.2.5** A-D; **4.1.1** A-I; **4.1.2** A-I; **4.2.1** A-D
	Web IntroComputing.org Piazza	**Formative Assessment: Creating Algorithms to Manipulate Images** Students write code to modify existing images using algorithms they create. They may use IntroComputing.org for examples if they wish. They post on Piazza their code and a screenshot of their images. They then provide feedback to their peers as Piazza comments and questions. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A, B, E; **1.2.2** A; **1.2.3** A; **1.2.4** A, B, F; **1.2.5** A-D; **4.1.1** A-I; **4.1.2** A-I; **4.2.1** A-D

Having students use Piazza to post their code and a screenshot of their manipulated images makes it easier for me to review their work. It also helps their peers to learn through commenting and posting questions.

EXPLORATION 4: EXPLORING ALGORITHMS

Estimated Time: 10 Hours

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms

Essential Understandings:
- EU 1.1, EU 1.2, EU 4.1, EU 4.2

Projects and Major Assignments:
- Creating Algorithms to Manipulate Images ▸ Huffman Coding Trees ▸ Comparing Algorithms for Image File Creation ▸ Designing Algorithms for Map Creation

Guiding Questions

▸ **What is an algorithm?** ▸ **What are common algorithms used for digital image manipulation?** ▸ **How can algorithms be evaluated for correctness and efficiency?** ▸ **How can messages be translated between multiple representations?** ▸ **How is map location data used to obtain practical information?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
4.1.2: Express an algorithm in a language. [P5] **4.2.4:** Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity. [P4]	**Web** Mr. Kick, "Resources"	**Instructional Activity: Huffman Coding Trees** I introduce students to Huffman coding trees and the algorithms associated with the creation and use of these binary trees through videos created by my former students. I create coding trees using specific input and discuss the correctness of those trees. Students replicate the process and apply that process to new data. Later in the course, during discussions about the transmission of information over the Internet, students enjoy not only using the trees to send messages that are generally not readable by the general public but also creating the trees and encoding their own messages. **Essential knowledge addressed: 4.1.2** A-I; **4.2.4** B-C
		Formative Assessment: Huffman Coding Trees I give students a frequency list for characters used to create messages. They independently use the frequencies to create a Huffman coding tree. Students review the work of their partner by comparing the tree they create with the one their partner created. Students discuss the similarities, differences, and correctness of the trees. **Essential knowledge addressed: 4.1.2** A-I; **4.2.4** B-C

Student work is assessed through comparison of the student-created tree and the correct tree associated with the given data. The number of nodes generated should match the original number of letters given in the problem. High-frequency letters should be higher in the tree than low-frequency letters.

EXPLORATION 4: EXPLORING ALGORITHMS

Estimated Time: 10 Hours

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms

Essential Understandings:
▶ EU 1.1, EU 1.2, EU 4.1, EU 4.2

Projects and Major Assignments:
▶ Creating Algorithms to Manipulate Images ▶ Huffman Coding Trees ▶ Comparing Algorithms for Image File Creation ▶ Designing Algorithms for Map Creation

Guiding Questions

▶ What is an algorithm? ▶ What are common algorithms used for digital image manipulation? ▶ How can algorithms be evaluated for correctness and efficiency? ▶ How can messages be translated between multiple representations? ▶ How is map location data used to obtain practical information?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
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] **LO 4.2.4:** Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity. [P4]	**Web** Mr. Kick, "Resources"	**Instructional Activity: Comparing Algorithms for Image File Creation** I introduce students to an algorithm used by a previous student in the course to generate an image file. I present a modified algorithm that generates the same image file much more quickly. I then ask students to compare the two algorithms in terms of the number of steps needed to complete each algorithm. Discussions related to these and other algorithms lead students to an understanding of how to evaluate algorithms based on both correctness and efficiency. **Essential knowledge addressed: 4.2.2** A-D; **4.2.3** A-C; **4.2.4** A-H
	Web Piazza	**Formative Assessment: Comparing Algorithms for Image File Creation** I give students pairs of algorithms designed to accomplish the same task in the form of Piazza polls. I ask them to independently select the best and worst of the two algorithms in terms of readability, memory usage, and runtime efficiency. I allow students to view the results of each poll after they submit their answers. Class discussions follow after all students have entered their choices in the polls. I encourage students to ask questions and contribute insights into why they selected their choices. Summaries of the ideas discussed in class are entered as replies to the Piazza polls. **Essential knowledge addressed: 4.2.2** A-D; **4.2.3** A-C; **4.2.4** A-H

Student choices are anonymous among peers but categorized by respondents when viewed by the instructor. Students who consistently answer correctly and post valuable insights into their reasoning score highest. Students who make errors but discuss their thinking and post corrections in their thinking also score highly. Students who post incorrect answers and do not post insights into appropriate modifications in their thinking score lowest.

EXPLORATION 4: EXPLORING ALGORITHMS

Estimated Time: 10 Hours

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms

Essential Understandings:
- EU 1.1, EU 1.2, EU 4.1, EU 4.2

Projects and Major Assignments:
- Creating Algorithms to Manipulate Images
- Huffman Coding Trees
- Comparing Algorithms for Image File Creation
- Designing Algorithms for Map Creation

Guiding Questions

- What is an algorithm?
- What are common algorithms used for digital image manipulation?
- How can algorithms be evaluated for correctness and efficiency?
- How can messages be translated between multiple representations?
- How is map location data used to obtain practical information?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
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] **LO 4.2.4:** Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity. [P4]	**Web** "Google Maps JavaScript API" JSFiddle	**Instructional Activity: Designing Algorithms for Map Creation** I show students the Google Maps JavaScript API. I then demonstrate past student function implementations and discuss their code. The examples include total distance traveled from a starting location to an ending location including a sequence of intermediate stopping points, the centroid of a collection of locations, and the animated display of a sequence of map locations. Students discuss in small groups, with each student speaking individually for 30 seconds, practical algorithms that they can implement using the API. Students implement the algorithms using JavaScript and the Google Maps API. They create the implementations using the JSFiddle website for HTML and JavaScript programming. **Essential knowledge addressed: 4.2.2** A-D; **4.2.3** A-C; **4.2.4** A-H
	Web JSFiddle	**Instructional Activity: Creating Map Websites** Students work in pairs in class to create websites on JSFiddle that use the features of Google Maps to illustrate travels that they have experienced or would like to experience. Each student Web page created with JSFiddle is associated with a URL that can be shared with anyone accessing the World Wide Web. It is popular for students to illustrate family vacations or travels associated with family relocations. Students have also animated sequences of locations taken from movies that they have seen. The animations often include text bubbles that provide insights into the movie plot corresponding to the film locations. Students share their work with the entire class at the end of the project by posting JSFiddle URLs on Piazza. **Essential knowledge addressed: 4.2.2** A-D; **4.2.3** A-C; **4.2.4** A-H

I encourage students to comment on both general and technical details associated with each website. Here I am looking for the general effectiveness of the website in presenting relevant information, the complexity of the algorithms implemented in the site, and the readability of the code and comments used to implement the algorithms.

EXPLORATION 4: EXPLORING ALGORITHMS

Estimated Time: 10 Hours

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms

Essential Understandings:
▶ EU 1.1, EU 1.2, EU 4.1, EU 4.2

Projects and Major Assignments:
▶ Creating Algorithms to Manipulate Images ▶ Huffman Coding Trees ▶ Comparing Algorithms for Image File Creation ▶ Designing Algorithms for Map Creation

Guiding Questions

▶ **What is an algorithm?** ▶ **What are common algorithms used for digital image manipulation?** ▶ **How can algorithms be evaluated for correctness and efficiency?** ▶ **How can messages be translated between multiple representations?** ▶ **How is map location data used to obtain practical information?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from this exploration are addressed.	**Web** Google Forms	**Summative Assessment: Exploring Algorithms** Students are given questions in a Google Form that are related to algorithms and their creation and use. They submit their answers to these multiple-choice and free-response questions to me within the form. **All of the exploration's essential knowledge statements are addressed.**

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

EXPLORATION 5: PROBLEM SOLVING WITH PROGRAMMING

Estimated Time: 20 Hours

BIG IDEA 2 Abstraction
BIG IDEA 5 Programming

Essential Understandings:
- EU 2.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:
- Baby Name Analysis
- Mad Libs
- Using Maps for Problem Solving

Guiding Questions

▶ How can programming be used to process data and obtain new information and knowledge? ▶ How can programs process user input to create amusing results? ▶ How can the Google Maps API be used to generate maps and solve map-related questions?

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 2.2.2:** Use multiple levels of abstraction to write programs. [P3] **LO 2.2.3:** Identify multiple levels of abstractions being used when writing programs. [P3] **LO 5.1.1:** Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. [P2] **LO 5.1.2:** Develop a correct program to solve problems. [P2] **LO 5.1.3:** Collaborate to develop a program. [P6] **LO 5.2.1:** Explain how programs implement algorithms. [P3] **LO 5.3.1:** Use abstraction to manage complexity in programs. [P3] **LO 5.4.1:** Evaluate the correctness of a program. [P4] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]	**Web** JSFiddle IntroComputing .org, "Table Data" "Top 10 Baby Names for 2014" W3Schools, "HTML"	**Instructional Activity: Baby Name Analysis** Students independently write JavaScript code to answer the questions posed about baby names (from the Social Security Administration database) in the "Table Data" section of IntroComputing.org. They also pose name-related questions and answer their own questions and the questions of their peers using JavaScript programming. This programming uses JSFiddle or text files that are loaded into a Web browser. Students who need to refresh their memory of JavaScript syntax and functionality may use the W3Schools website's "HTML" tutorial. Students are required to explain — in small peer groups and in their weekly journal, using appropriate terminology and various levels of abstractions — how they solved the problems. **Essential knowledge addressed: 2.2.1** A-C; **2.2.2** A, B; **2.2.3** K; **5.1.1** A-F; **5.1.2** A-J; **5.1.3** A-F; **5.2.1** A-K; **5.3.1** A-N; **5.4.1** A-O; **5.5.1** E-J

Students enjoy exploring names that are meaningful to them — usually family names or names of their friends. Exploring changes in the rankings of those names over decades is particularly appealing to most students.

EXPLORATION 5: PROBLEM SOLVING WITH PROGRAMMING

Estimated Time: 20 Hours

BIG IDEA 2 Abstraction
BIG IDEA 5 Programming

Essential Understandings:
- EU 2.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:
- Baby Name Analysis
- Mad Libs
- Using Maps for Problem Solving

Guiding Questions

▸ How can programming be used to process data and obtain new information and knowledge? ▸ How can programs process user input to create amusing results? ▸ How can the Google Maps API be used to generate maps and solve map-related questions?

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 2.2.2:** Use multiple levels of abstraction to write programs. [P3] **LO 2.2.3:** Identify multiple levels of abstractions being used when writing programs. [P3] **LO 5.1.1:** Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. [P2] **LO 5.1.2:** Develop a correct program to solve problems. [P2] **LO 5.1.3:** Collaborate to develop a program. [P6] **LO 5.2.1:** Explain how programs implement algorithms. [P3] **LO 5.3.1:** Use abstraction to manage complexity in programs. [P3] **LO 5.4.1:** Evaluate the correctness of a program. [P4] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]	**Web** Google Docs journals IntroComputing .org JSFiddle	**Formative Assessment: Problem Solving Through Programming** Students work in pairs to write programs that solve problems posed on IntroComputing.org. They write all the solutions for these problems in the text areas provided within the site. Students also must pose problems to be solved by their table partners. Students are allowed to write code that solves problems posed by their peers using JSFiddle or simple text files that are loaded into the Web browser. **Essential knowledge addressed: 2.2.1** A-C; **2.2.2** A, B; **2.2.3** K; **5.1.1** A-F; **5.1.2** A-J; **5.1.3** A-F; **5.2.1** A-K; **5.3.1** A-N; **5.4.1** A-O; **5.5.1** E-J

Students present the problem of their choice and their programming solution to their peers and in their weekly journals. It is important to emphasize to students that JavaScript is run within a Web browser. Sites like IntroComputing.org and JSFiddle provide relatively easy, low-overhead methods for entering and running the code. JavaScript code also can be written in text files that are formatted as HTML files with embedded JavaScript.

EXPLORATION 5: PROBLEM SOLVING WITH PROGRAMMING

Estimated Time: 20 Hours

BIG IDEA 2 Abstraction
BIG IDEA 5 Programming

Essential Understandings:
- EU 2.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:
- Baby Name Analysis
- Mad Libs
- Using Maps for Problem Solving

Guiding Questions

▸ How can programming be used to process data and obtain new information and knowledge? ▸ How can programs process user input to create amusing results? ▸ How can the Google Maps API be used to generate maps and solve map-related questions?

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 2.2.2:** Use multiple levels of abstraction to write programs. [P3] **LO 5.1.1:** Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. [P2] **LO 5.1.2:** Develop a correct program to solve problems. [P2] **LO 5.1.3:** Collaborate to develop a program. [P6] **LO 5.2.1:** Explain how programs implement algorithms. [P3] **LO 5.3.1:** Use abstraction to manage complexity in programs. [P3] **LO 5.4.1:** Evaluate the correctness of a program. [P4] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]	**Web** JSFiddle Piazza	**Instructional Activity: Mad Libs** Students work in groups of four and create a story about a topic of interest. They then must create a Web page for the story that contains images, colors, and general setting descriptions of their choosing. They must include input devices (e.g., text box, text area, checkbox, radio button, pull-down menu) through which users will input nouns and verbs to be used in the story as determined by the group. Students brainstorm about how to divide the programming task among the group in order to use the strengths of each individual. They must write a JavaScript program within the Web page that uses input from the page to determine the final details of their story. Stories are shared using JSFiddle URLs posted on Piazza. **Essential knowledge addressed:** **2.2.1** A-C; **2.2.2** A-B; **5.1.1** A-F; **5.1.2** A-J; **5.1.3** A-F; **5.2.1** A-K; **5.3.1** A-O; **5.4.1** A-N; **5.5.1** E-J

EXPLORATION 5: PROBLEM SOLVING WITH PROGRAMMING

Estimated Time: 20 Hours

BIG IDEA 2 Abstraction
BIG IDEA 5 Programming

Essential Understandings:
- EU 2.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:
- Baby Name Analysis
- Mad Libs
- Using Maps for Problem Solving

Guiding Questions

- How can programming be used to process data and obtain new information and knowledge?
- How can programs process user input to create amusing results?
- How can the Google Maps API be used to generate maps and solve map-related questions?

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 2.2.2:** Use multiple levels of abstraction to write programs. [P3] **LO 5.1.1:** Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. [P2] **LO 5.1.2:** Develop a correct program to solve problems. [P2] **LO 5.1.3:** Collaborate to develop a program. [P6] **LO 5.2.1:** Explain how programs implement algorithms. [P3] **LO 5.3.1:** Use abstraction to manage complexity in programs. [P3] **LO 5.4.1:** Evaluate the correctness of a program. [P4] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]	**Web** JSFiddle	**Formative Assessment: Using User Input** I give students, who are working in pairs, sets of input and ask them to write JavaScript code that processes the input and produces the desired output. For example, numerical data in the form of coordinate pairs may be used for computations such as distances between points. String data may be used for alphabetical comparisons or word searches. Tag data in the form of text may be used for Web page output formatting. All data is input through the use of Web page input elements, processed using JavaScript, and output in the form of Web page content. **Essential knowledge addressed: 2.2.1** A-C; **2.2.2** A-B; **5.1.1** A-F; **5.1.2** A-J; **5.1.3** A-F; **5.2.1** A-K; **5.3.1** A-O; **5.4.1** A-N; **5.5.1** E-J	*Students share their Web pages with me using JSFiddle.*
	Web "Google Maps JavaScript API" JSFiddle	**Instructional Activity: Using Maps for Problem Solving** I introduce a sample Web page that includes a JavaScript program that allows for user input and displays output in the form of map data. One portion of the page allows the user to enter addresses, one per line of text. Clicking the submit button runs the JavaScript program that processes the addresses, generates map coordinates, and plots markers on the map at each address. In pairs, students create a Web page that contains a map displaying a region of interest. The page includes input devices that allow the user to enter data associated with map manipulations. Each student writes a JavaScript program within the Web page that uses input to manipulate the map in creative ways. Web pages are shared with the class using JSFiddle. **Essential knowledge addressed: 2.2.1** A-C; **2.2.2** A-B; **5.1.1** A-F; **5.1.2** A-J; **5.1.3** A-F; **5.2.1** A-K; **5.3.1** A-O; **5.4.1** A-N; **5.5.1** E-J	*Input devices in the Web page should include text box, text area, checkbox, radio button, pull-down menu, and so on. Some ways students can manipulate the map are by adding graphics, computing distances between locations, and estimating travel times between those locations.*

EXPLORATION 5: PROBLEM SOLVING WITH PROGRAMMING

Estimated Time: 20 Hours

BIG IDEA 2 Abstraction
BIG IDEA 5 Programming

Essential Understandings:
- EU 2.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:
- Baby Name Analysis
- Mad Libs
- Using Maps for Problem Solving

Guiding Questions

- **How can programming be used to process data and obtain new information and knowledge?**
- **How can programs process user input to create amusing results?**
- **How can the Google Maps API be used to generate maps and solve map-related questions?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from this exploration are addressed.	**Web** Google Forms IntroComputing .org	**Summative Assessment: Problem Solving With Programming** I give students a set of data (such as data from IntroComputing .org) and questions about that data. They must individually write programs that answer the given questions by effectively processing the various forms of data provided. Their code must be appropriately commented and accompanied by a general explanation of how the program accomplished its tasks. Students work on the programming problems using the programming tools of their choice. They write code that answers the given questions and then submit their answer using a Google Form. Students must also select one well-commented program that solves one of the given problems and submit the code for that program as text in the same Google Form. **All of the exploration's essential knowledge statements are addressed.**

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

EXPLORATION 6: GUIDED INTERNET EXPLORATIONS

Estimated Time: 15 Hours

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

Essential Understandings:

- EU 1.1, EU 1.2, EU 2.1, EU, 2.2, EU 2.3, EU 6.1, EU 6.2, EU 6.3

Projects and Major Assignments:

- Simulating Internet Communications
- Understanding the Internet
- The Hardware and Binary Connection
- Creating Videos Explaining Characteristics of the Internet

Guiding Questions

- **How does communication over the Internet occur?** ▸ **What resources are available for better understanding the Internet?** ▸ **What are the hardware components associated with the Internet and its functionality?** ▸ **What are the characteristics of the Internet that make it so powerful and flexible?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
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] **LO 2.3.1:** Use models and simulations to represent phenomena. [P3] **LO 2.3.2:** Use models and simulations to formulate, refine, and test hypotheses. [P3] **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]	**Web** Piazza YouTube	**Instructional Activity: Simulating Internet Communications** I ask students to search YouTube using the keywords *Huffman coding tree encoding* and to select videos that effectively present encoding and decoding using Huffman coding trees. Each group posts the URL of its video choice on Piazza. I build a Huffman coding tree based on sample letter frequencies from instructor- and student-generated messages. I translate the messages into binary form using large (1) and small (0) paper clips linked together in fixed-length packets. I label the packets with destination information that ensures reassembly after transmission. Packets are passed through student "routers," who receive packets and pass them toward their destination. Packets are reassembled at the message destination and translated from binary to the original message form. Students enjoy sending "secret messages" to their peers. **Essential knowledge addressed: 2.1.1** A-E; **2.1.2** D-F; **2.3.1** A-D; **2.3.2** A-H; **6.1.1** A-I; **6.2.1** A-D	*To explain the historical need for efficient message representations in binary code, I present an Apple II1 computer with 64KB of RAM. I share stories of current software engineers who create compression schemes to more rapidly store and transmit the content of chips during program failure. Be sure to emphasize that Huffman coding is not cryptography but that it can be used to obscure the meaning of messages for the general public.*
	Web Google Docs journals	**Formative Assessment: Simulating Internet Communications** Students work in small table groups to create their own messages. They create, on paper, a Huffman coding tree for their messages using the character frequencies for all of their messages combined. Each group uses its tree to encode its message into binary form, breaks the messages into packets, and sends its packets through designated "routers" to an intended destination. If the receiving group, when given the coding tree, decodes the message, the transmission is considered a success. Students summarize their experiences in their weekly journal on Google Docs, where I can review and provide feedback as needed. **Essential knowledge addressed: 2.1.1** A-E; **2.1.2** D-F; **2.3.1** A-D; **2.3.2** A-H; **6.1.1** A-I; **6.2.1** A-D	*The student journals must include an example of a message that was sent, the tree used for encoding and decoding, and a summary of the difficulties and successes experienced during the entire process.*

EXPLORATION 6: GUIDED INTERNET EXPLORATIONS

Estimated Time: 15 Hours

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

Essential Understandings:
- EU 1.1, EU 1.2, EU 2.1, EU, 2.2, EU 2.3, EU 6.1, EU 6.2, EU 6.3

Projects and Major Assignments:
- Simulating Internet Communications
- Understanding the Internet
- The Hardware and Binary Connection
- Creating Videos Explaining Characteristics of the Internet

Guiding Questions

▶ **How does communication over the Internet occur?** ▶ **What resources are available for better understanding the Internet?** ▶ **What are the hardware components associated with the Internet and its functionality?** ▶ **What are the characteristics of the Internet that make it so powerful and flexible?**

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 6.2.1:** 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 6.3.1:** Identify existing cybersecurity concerns and potential options that address these issues with the Internet and the systems built on it. [P1]	**Print** Abelson, Ledeen, and Lewis, chapter 7 and Appendix **Web** Internet Archive Way Back Machine, "Newbury Park High School" Archives "Resources," Mr. Kick	**Instructional Activity: Understanding the Internet** I post URLs associated with different iterations of the school website over time on Piazza. I encourage students to view the history of the site and consider the persistent nature of data on the Internet. Students gather in table groups and generate a set of questions per table that they have about the Internet. Each table posts its questions on Piazza (one post per unique question). Students use the Internet, chapter 7 and the appendix of *Blown to Bits,* and the resources listed on my course website to research the answers to their peers' questions. Each table discusses and posts its best responses. Students respond to one another's comments, refining the answers and eventually resulting in fairly accurate responses to the original questions. **Essential knowledge addressed: 6.1.1** A-I; **6.2.1** A-D; **6.2.2** A-K; **6.3.1** A-M	*I monitor the posts and contribute to the process as needed. I add important questions that are not asked by students to Piazza as well. Generally, students do very well at obtaining and summarizing answers to the questions posted by their peers.*
LO 2.2.3: Identify multiple levels of abstractions that are used when writing programs. [P3]	**Web** IntroComputing .org, "Computer Hardware" American Computer Science League, "Sample Problems"	**Instructional Activity: The Hardware and Binary Connection** Students read the "Computer Hardware" page on IntroComputing. org and explore the concepts (e.g., hardware, chip, CPU, transistor) and images (pictures of hardware) presented while examining actual hardware in the classroom (I use an Apple II1 computer). Students discuss the topics in table groups and answer questions about the hardware I present to them. I introduce competition-style questions similar to those found on the "Sample Problems" page on the American Computer Science League site on the topics of Boolean algebra and logic gates. Sample problems are presented, discussed, and solved in large groups, small groups, and individually. Students make connections between the hardware used to create computing technology and binary systems. **Essential knowledge addressed: 2.2.3** E-K	

EXPLORATION 6: GUIDED INTERNET EXPLORATIONS

Estimated Time: 15 Hours

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

Essential Understandings:

- EU 1.1, EU 1.2, EU 2.1, EU, 2.2, EU 2.3, EU 6.1, EU 6.2, EU 6.3

Projects and Major Assignments:

- Simulating Internet Communications
- Understanding the Internet
- The Hardware and Binary Connection
- Creating Videos Explaining Characteristics of the Internet

Guiding Questions

- **How does communication over the Internet occur?**
- **What resources are available for better understanding the Internet?**
- **What are the hardware components associated with the Internet and its functionality?**
- **What are the characteristics of the Internet that make it so powerful and flexible?**

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** Google Docs journals Piazza	**Formative Assessment: The Hardware and Binary Connection** Students summarize the concepts related to logic gates and binary representations on Piazza in small- and large-group discussions and in their weekly journal. They also answer competition-style questions associated with Boolean algebra and logic gates. **Essential knowledge addressed: 2.2.3** E-K	*I facilitate games in which students work together in small teams to respond to Boolean algebra and logic gate questions.*
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6]	**Web** Piazza	**Instructional Activity: Creating Videos Explaining Characteristics of the Internet** I list the major characteristics of the Internet on the board (hierarchical, redundant, fault tolerant, standardized through protocols, packet-switched, and scalable), and students vote using a Piazza poll for the characteristic that is most interesting to them. After participating in the poll, students move to a location in the room designated as representing the characteristic they selected and divide into teams. I instruct students to create a video of no longer than one minute that illustrates the characteristic of the Internet that they voted for. When each team has completed its video, I (or a student volunteer) edit all submissions into a single video illustrating the major characteristics of the Internet. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A, B, E; **1.2.2** A; **1.2.3** A; **1.2.4** A, B, F	*Effective collaboration is essential for success in this project. I may make adjustments in team numbers after initial voting takes place to ensure equity in the workload for each student. Video production done well is a big job and requires planning, scripting, staging, video capturing, editing, sound management, and so on. Each of these tasks can be a large job. Encourage students to divide up the work when possible to ensure higher-quality work in a relatively short time.*

EXPLORATION 6: GUIDED INTERNET EXPLORATIONS

Estimated Time: 15 Hours

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

Essential Understandings:

- EU 1.1, EU 1.2, EU 2.1, EU, 2.2, EU 2.3, EU 6.1, EU 6.2, EU 6.3

Projects and Major Assignments:

- Simulating Internet Communications
- Understanding the Internet
- The Hardware and Binary Connection
- Creating Videos Explaining Characteristics of the Internet

Guiding Questions

▸ **How does communication over the Internet occur?** ▸ **What resources are available for better understanding the Internet?** ▸ **What are the hardware components associated with the Internet and its functionality?** ▸ **What are the characteristics of the Internet that make it so powerful and flexible?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from this exploration are addressed.	**Web** Google Forms	**Summative Assessment: Guided Internet Explorations** I give students questions related to the Internet in the form of multiple-choice and free-response questions using Google Forms. Students submit their answers, which I accumulate into a single spreadsheet for relatively quick assessment. Students are not only able to take an online assessment about the Internet using the Internet, but they also see the power of the Internet for effective dissemination of information, accumulation of responses from relatively large numbers of people, and rapid assessment of those responses through programming. **All of the exploration's essential knowledge statements are addressed.**

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

EXPLORATION 7: IDENTIFYING INNOVATIONS

Estimated Time: 20 Hours

BIG IDEA 7 Global Impact

Essential Understandings:
- EU 7.1, EU 7.2, EU 7.3, EU 7.4, EU 7.5

Projects and Major Assignments:
- Identifying and Summarizing Articles About Computing Innovations and Impacts
- Wharton Top 30 Innovations List
- Debating the Beneficial and Harmful Impacts of Computing on Our Society

Guiding Questions

- What computing innovations have the greatest impact on your life?
- What recent computing innovations are on the Wharton Top 30 Innovations list?
- What are the beneficial and harmful effects of computing on your life?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 7.1.1: Explain how computing innovations affect communication, interaction, and cognition. [P4] **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] **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] **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** ACM TechNews Piazza Mr. Kick, "Resources"	**Instructional Activity: Identifying and Summarizing Articles About Computing Innovations and Impacts** Students go to ACM TechNews and select one article that identifies a computing innovation that affects or has the potential to affect our society. They then write a summary of the article on Piazza and are given an opportunity to verbally share their insights with their table partners and the entire class. Their summary must include a clear statement of how the innovation functions, how it is used and by whom or what, and how it affects people and society (both beneficial and harmful impacts). Students post their summaries to Piazza. **Essential knowledge addressed: 7.1.1** A-I, M-O; **7.1.2** A-G; **7.2.1** A-D; **7.3.1** A-C, E, G, J, M-O; **7.4.1** A-D; **7.5.1** A-C; **7.5.2** A, B
	Web Piazza	**Formative Assessment: Discussing and Summarizing Ideas from Readings** Students read the innovation article summary written by a classmate whom I randomly selected as their partner. They must ask the summary author clarifying questions about the innovation. They then write the insights gained through their questioning and discussions as a response to the original article summary posted on Piazza. **Essential knowledge addressed: 7.1.1** A-I, M-O; **7.1.2** A-G; **7.2.1** A-D; **7.3.1** A-C, E, G, J, M-O; **7.4.1** A-D; **7.5.1** A-C; **7.5.2** A, B

ACM TechNews is a constant source of new information and cutting-edge innovations being explored and developed by computer scientists today. Whereas students have previously read articles in this course based on their general interests, their previous experiences now allow them to be more selective in their choices of articles.

Writing about innovations described in ACM TechNews helps prepare students to successfully complete the Explore — Impact of Computing Innovations Performance Task. Having the opportunity to question another student about his or her article summary helps both students improve the quality of their writing and expand the level of detail provided in their writing.

EXPLORATION 7: IDENTIFYING INNOVATIONS

Estimated Time: 20 Hours

BIG IDEA 7 Global Impact

Essential Understandings:

- EU 7.1, EU 7.2, EU 7.3, EU 7.4, EU 7.5

Projects and Major Assignments:

- Identifying and Summarizing Articles About Computing Innovations and Impacts
- Wharton Top 30 Innovations List
- Debating the Beneficial and Harmful Impacts of Computing on Our Society

Guiding Questions

- **What computing innovations have the greatest impact on your life?**
- **What recent computing innovations are on the Wharton Top 30 Innovations list?**
- **What are the beneficial and harmful effects of computing on your life?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
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 7.5.1:** Access, manage, and attribute information using effective strategies. [P1] **LO 7.5.2:** Evaluate online and print sources for appropriateness and credibility. [P5]	**Web** Piazza "A World Transformed: What Are the Top 30 Innovations of the Last 30 Years?"	**Instructional Activity: Wharton Top 30 Innovations List** I give students time to read, in pairs, the list of the top 30 innovations from Wharton and to select two computing innovations that most interest them. Students then write a technical summary, as a Piazza post, of how computing is related to those innovations. Finally, I give students an opportunity to share their insights with their table partners and the entire class. **Essential knowledge addressed: 7.1.1** A-I, L-N; **7.2.1** A-C; **7.5.1** A-C; **7.5.2** A, B
	Web Google Docs	**Formative Assessment: Wharton Top 30 Innovations List** Students summarize the technical aspects of a commonly known computing innovation that I select for them. I provide class time for students to research the computing innovation in small table groups. They then create a collaborative summary of the technical aspects of the computing innovation through the use of Google Docs. Each table presents its summary to the rest of the class. Peers are encouraged to ask questions to help students provide details that they omitted in their initial summary. **Essential knowledge addressed: 7.1.1** A-I, L-N; **7.2.1** A-C; **7.5.1** A-C; **7.5.2** A, B

Technical summaries associated with computing are usually relatively difficult for students to write. Emphasize to students that they should share their writing and use questions about their writing from their peers as an opportunity to make revisions, correct errors, and clarify ambiguities.

EXPLORATION 7: IDENTIFYING INNOVATIONS

Estimated Time: 20 Hours

BIG IDEA 7 Global Impact

Essential Understandings:

- EU 7.1, EU 7.2, EU 7.3, EU 7.4, EU 7.5

Projects and Major Assignments:

- Identifying and Summarizing Articles About Computing Innovations and Impacts
- Wharton Top 30 Innovations List
- Debating the Beneficial and Harmful Impacts of Computing on Our Society

Guiding Questions

- **What computing innovations have the greatest impact on your life?**
- **What recent computing innovations are on the Wharton Top 30 Innovations list?**
- **What are the beneficial and harmful effects of computing on your life?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 7.1.1: Explain how computing innovations affect communication, interaction, and cognition. [P4] **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] **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] **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: Debating the Beneficial and Harmful Impacts of Computing on Our Society** I ask students to stand and move to a portion of the room that indicates whether they believe computing has provided more benefits to society than harm. Another portion of the room is designated for students who disagree. I conduct a question-and-answer session that helps shift students to balance groups, if needed. I then give students time in class to collaborate with their group, plan arguments for their side, and debate the topic. I record the debate and make a video that summarizes it. Students earn extra credit by volunteering to edit the video. **Essential knowledge addressed: 7.1.1** A-O; **7.1.2** A-G; **7.2.1** A-G; **7.3.1** A-Q; **7.4.1** A-E; **7.5.1** A-C; **7.5.2** A, B	*Students traditionally enjoy this activity. It gives them the opportunity to express their opinions and logically support them with evidence. Students learn from their individual research but also from the research and discussions with their peers.*
	Web Google Docs journals	**Formative Assessment: Debating the Beneficial and Harmful Impacts of Computing on Our Society** Students summarize in their journals their perspectives about the computing debate after they have listened to all arguments that were made. They must include a summary of their position before the debate, a list of points made during the debate that most affected their perspective, and a summary of their position after the conclusion of the debate. **Essential knowledge addressed: 7.1.1** A-O; **7.1.2** A-G; **7.2.1** A-G; **7.3.1** A-Q; **7.4.1** A-E; **7.5.1** A-C; **7.5.2** A, B	*The process of summarizing their perspectives in their journal helps students better understand their own thinking. Often, they are not fully aware of their own opinions until they have to take a side and write about their reasoning.*

EXPLORATION 7: IDENTIFYING INNOVATIONS

Estimated Time: 20 Hours

BIG IDEA 7 Global Impact

Essential Understandings:
- EU 7.1, EU 7.2, EU 7.3, EU 7.4, EU 7.5

Projects and Major Assignments:
- Identifying and Summarizing Articles About Computing Innovations and Impacts
- Wharton Top 30 Innovations List
- Debating the Beneficial and Harmful Impacts of Computing on Our Society

Guiding Questions

- **What computing innovations have the greatest impact on your life?**
- **What recent computing innovations are on the Wharton Top 30 Innovations list?**
- **What are the beneficial and harmful effects of computing on your life?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 7.1.1: Explain how computing innovations affect communication, interaction, and cognition. [P4] **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] **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]	**Print** Abelson, Ledeen, and Lewis, Conclusion **Web** Google Docs	**Instructional Activity: Reading and Discussing the Conclusion to *Blown to Bits*** I assign the conclusion to *Blown to Bits* as an in-class reading assignment. I ask each table to identify the technologies and social settings mentioned in the text and to use that information to write, in a shared Google Doc, the time period in which the book was researched and written. In that same Google Doc, students should list technologies and settings that might have been used if the book had been written today. Each group presents one of the technologies it listed and one of the social settings it identified. Groups take turns sharing their ideas until all groups have been able to share all of the technologies and social settings they have identified. **Essential knowledge addressed: 7.1.1** A-O; **7.1.2** A-G; **7.2.1** A-G; **7.3.1** A-Q; **7.5.1** A-C; **7.5.2** A, B
All of the learning objectives from this exploration are addressed.	**Web** Google Forms	**Summative Assessment: Identifying Innovations** I give students questions in a Google Form that are related to innovations and how they are connected to people and society. Students submit their answers to these multiple-choice and free-response questions to me within the form. **All of the exploration's essential knowledge statements are addressed.**

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

AP Performance Task

ESTIMATED TIME: 8 hours

Explore – Impact of Computing Innovations

Students complete the Explore – Impact of Computing Innovations Performance Task toward the end of the second semester of the course. They have already debated the effects of computing on society and have written several entries about the impact of computing as related to personally interesting topics. They have received both informal and formal feedback from the instructor and their peers on their writing and its effectiveness in conveying their ideas on these topics. All of the exploration themes have been at least partially completed at this time in the course. Students are ready to individually complete the performance task at this time.

AP Performance Task

ESTIMATED TIME: 12 hours

Create – Applications from Ideas

Students complete the Create – Applications from Ideas Performance Task as their final task in the course before taking the AP Computer Science Principles Exam. At this point in the course, students are thoroughly familiar with JavaScript programming. They generally receive very positive feedback on their efforts at producing computational art for the all-school art show, and they are highly motivated to continue the creation of personally relevant artifacts. They have collaboratively worked almost an entire school year with their peers. At this stage in the course, they are comfortable creating and summarizing their work with partners.

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

Abelson, Hal, Ken Ledeen, and Harry Lewis. *Blown to Bits: Your Life, Liberty, and Happiness After the Digital Explosion.* New York: Addison-Wesley, 2008. Accessed June 15, 2015. **http://www.bitsbook.com**.

ACM TechNews. Accessed July 14, 2015. **http://technews.acm.org**.

Fathom Dynamic Data Software. Accessed July 22, 2015. **http://fathom.concord.org**.

Google Docs. Accessed July 14, 2015. **https://docs.google.com**.

Google Forms. Accessed July 14, 2015. **https://www.google.com/forms/about**.

"Google Maps JavaScript API." Google Developers. Accessed July 20, 2015. **https://developers.google.com/maps/documentation/javascript/examples**.

IntroComputing.org. Accessed July 14, 2015. **http://introcomputing.org**.

JSFiddle. Accessed July 28, 2015. **http://JSFiddle.net**.

Kick, Richard. "Webpage Programming with JavaScript." Accessed July 22, 2015. **http://kickstyle.net/~rkick/courses/apcs/principles**.

Piazza. Accessed July 14, 2015. **http://piazza.com**.

"Resources." Mr. Kick. Accessed July 14, 2015. **https://sites.google.com/site/nphsrkick/Home/csprinciples/resources**.

Exploration 1 (Creativity and Computing) Resources

Paul, Jody, Rich Kick, and Paul Tymann. "AP CSP Curriculum Module — Creativity and Global Impact." College Board. Accessed September 22, 2015. **https://apcommunity.collegeboard.org/group/csprinciples/resource-details?p_p_id=contentItem_WAR_aptccontentitemportlet&p_p_lifecycle=0&p_p_col_id=column-1&p_p_col_count=1&p_r_p_1243656882_resourceId=47534342&pageQueryString=p_l_id%3D44665487%26p_v_l_s_g_id%3D0%26p_p_id%3DSearch_WAR_aptcsearchportlet%26_Search_WAR_aptcsearchportlet_pageName%3DresourceLibrary%26backToSearch%3Dtrue**.

"Blockly Games: Turtle." Blockly Games. Accessed July 28, 2015. **https://blockly-games.appspot.com/turtle?lang=en**.

Mr. doob. Accessed July 14, 2015. **http://mrdoob.com**.

Pencil Code. Accessed July 15, 2015. **https://pencilcode.net**.

Recursive Drawing. Accessed July 14, 2015. **http://recursivedrawing.com**.

"Rihanna — Diamonds (Live on SNL)." YouTube. Video, 4:07. Accessed July 14, 2015. **https://www.youtube.com/watch?v=2LT23ixDaJo**.

Exploration 2 (Identifying and Using Abstractions) Resources

"The Art of Abstraction – Computerphile." Computerphile. YouTube. Video, 5:21. Accessed July 8, 2015. **https://www.youtube.com/watch?v=p7nGcY73epw**.

"YouTube Data API: JavaScript Code Samples." Google Developers. Accessed July 14, 2015. **https://developers.google.com/youtube/v3/code_samples/javascript**.

Exploration 3 (Using Data) Resources

Google Drive. Accessed July 14, 2015. **https://drive.google.com**.

Exploration 4 (Exploring Algorithms) Resources

"Algorithms." Google Inside Search. Accessed July 14, 2015. **http://www.google.com/insidesearch/howsearchworks/algorithms.html**.

"How Search Works: From Algorithms to Answers" Google Inside Search. Accessed July 14, 2015. **http://www.google.com/insidesearch/howsearchworks/thestory/index.html**.

Exploration 5 (Problem Solving with Programming) Resources

"HTML." W3Schools. Accessed July 15, 2015. **http://www.w3schools.com**.

"Table Data." IntroComputing.org. Accessed July 14, 2015. **http://introcomputing.org**.

"Top 10 Baby Names for 2014." Social Security Administration. Accessed July 15, 2015. **http://www.ssa.gov/oact/babynames**.

Exploration 6 (Guided Internet Explorations) Resources

"Computer Hardware." IntroComputing.org. Accessed August 11, 2015. **http://introcomputing.org/hardware-1.html**.

Newbury Park High School. Accessed July 14, 2015. **http://nphs.org**.

"Newbury Park High School — June 12, 1998." Internet Archive Way Back Machine. Accessed July 14, 2015. **http://web.archive.org/web/19980207072056/http://www.nphs.org**.

"Newbury Park High School — March 1, 2000." Internet Archive Way Back Machine. Accessed July 14, 2015. **http://web.archive.org/web/20000301052305/http://www.nphs.org**.

Resources (continued)

"Newbury Park High School — July 22, 2001." Internet Archive Way Back Machine. Accessed July 14, 2015. **http://web.archive.org/web/20010722013253/http://nphs.org**.

"Newbury Park High School — October 1, 2003." Internet Archive Way Back Machine. Accessed July 14, 2015. **http://web.archive.org/web/20031001133739/http://www.nphs.org**.

"Newbury Park High School — December 14, 2007." Internet Archive Way Back Machine. Accessed July 14, 2015. **http://web.archive.org/web/20071214154949/http://www.nphs.org**.

"Newbury Park High School — October 6, 2008." Internet Archive Way Back Machine. Accessed July 14, 2015. **http://web.archive.org/web/20081006040719/http://www.nphs.org**.

"Sample Problems." American Computer Science League. Accessed August 11, 2015. **http://www.acsl.org/samples.htm**.

YouTube. Accessed July 21, 2015. **http://www.youtube.com**.

Exploration 7 (Identifying Innovations) Resources

"A World Transformed: What Are the Top 30 Innovations of the Last 30 Years?" Knowledge@Wharton. Accessed July 14, 2015. **http://knowledge.wharton.upenn.edu/article/a-world-transformed-what-are-the-top-30-innovations-of-the-last-30-years**.