TAMALPAIS UNION HIGH SCHOOL DISTRICT
LARKSPUR, CA

COURSE OF STUDY

ADVANCED PLACEMENT COMPUTER SCIENCE PRINCIPLES

Course Author: Project Lead the Way/Kelly Kennedy
School: Tamalpais High School
Length of Course: 1 Year
Subject Area and Discipline: Applied Technology, Computer Science
Grade Levels: 10-12.
Prerequisites: Computer Programming 1/2 or instructor approval
Are you seeking UC approval? Yes, as a g elective
Requirements satisfied: AP College Board credit, UC "g" elective credit

Introduction

This course focuses on the principles of computer science and technology, and is a project-based AP course designed to engage upper-class students who want a more self-directed approach to programming. Using the Python programming language as a primary tool, the course aims to develop computational thinking skills and to generate excitement about career paths that utilize computing. Projects and activities include interactive story and game development, app development with graphical user interfaces (GUIs), visualization of data, data simulation/processing, networking, and cybersecurity. Students develop programming expertise and explore information technology (IT) and the Internet. In addition to a multiple-choice component, the AP exam for this course requires students to submit two culminating performance tasks: a "create task," which is a programming project designed in a language of their choice and an "exploratory task," focused on researching and evaluating an innovation and the data it generates or affects on our society.

Structured activities in this course of study slowly progress to open-ended projects, and student-driven problems that require planning, documentation, communication, and other professional skills. Problems aim for ground-level entry with no ceiling: all students can successfully engage in the problems while students showing greater achievement can be challenged to work further. The units and activities are designed to address the five primary course objectives of AP Computer Science Principles:

- To develop problem solving and computational thinking skills
- To generate excitement about the field of computing
- To introduce computational tools that foster creativity
- To build awareness of career opportunities in all fields for people with computational skills
- To consider issues raised by the present and future societal impact of computing

Course Content

It’s important to consider that many units in this course can be rearranged across the year or modified to adjust to an alternative language of focus. Python can be substituted with Java or JavaScript, and the instructor can focus on more traditional programming languages as an alternative to the first two units covering Scratch and MIT App Inventor.

Unit 1.1 Algorithms and Agile Development

The goal of this unit is for students to discover the fun of creating an algorithm to accomplish a task. Students use a Scratch algorithm to manipulate images and audio recordings of themselves and learn about the common roles played by variables in a program. Object-oriented concepts are introduced using a variation of Scratch’s “sprites” and broadcasts. Students learn about state diagrams and control flow using the Lightbot video game and conclude the unit by designing and creating their own Scratch video game using pair programming. The software design process, from specification to user testing, is introduced as part of the concluding project.

Understandings

- U1.01 - Computing fosters creative expression, sometimes resulting in artifacts (K1.01) (S1.01, S8.12)
- U1.03 - Programming is a creative endeavor (S5.07, S5.01)
- U4.01 - Programs implement algorithms to solve problems (K4.05, K4.01)
- U4.02 - Algorithms can be analyzed for efficiency, and appropriate algorithms can be selected based upon efficiency
- U4.03 - Empirical analysis of algorithms requires a systematic approach (S5.09, S5.13)
- U5.02 - Solutions in a programming language are created by breaking a problem apart into component problems (S5.01, S4.01)
- U5.03 - Creating solutions with computation requires a persistent, iterative problem-solving approach (K5.01) (S5.09, S5.07)
- U5.04 - Programming requires an understanding of mathematical operations and data abstractions (K4.05)
- U5.06 - Functions with arguments make code modular and reusable (S5.01)
- U5.07 - Programmers create high-level documentation to communicate the purpose and function of their code (S8.02)
- U7.01 - Mobile and networked computing have transformed commerce, social interactions, news sourcing and dissemination, and culture (K7.02, K7.11, K7.13) (S8.07)
- U7.05 - New opportunities for human creativity and innovation exist because of networked, mobile, and embedded computing (K1.01)
- U8.01 - Computing artifacts and programs can be higher quality as a result of collaboration (S8.08, S8.09, S8.05, S8.01)
- U8.02 - Working in a team requires effective communication, clear responsibilities, and attention to interpersonal relationships (S8.09, S8.04, S8.05)
- U8.04 - Creative ideas and technical solutions must be communicated in a clear and concise manner (S8.12)
- U8.05 - How people present themselves affects how their work is received (S8.13, S8.12)

- U9.01 - Computer science and information technology careers offer creative job opportunities for individuals with a wide variety of backgrounds and goals (K9.04)
- U9.03 - Computational thinking boosts most career paths (K9.04, K9.06)

Knowledge

Students will:

- K1.01 - Describe the role of creativity in designing an attractive, functional, and accessible graphical user interface (U1.01, U7.05)
- K4.01 - Describe a computer as responding to input in a deterministic manner that depends only on input and on the computer's state (U4.01)
- K4.05 - Recognize common patterns employing variables, including value accumulation, list aggregation, and iteration across the elements of a collection (U4.01, U5.04)
- K5.01 - Describe ways to identify the existence and location of errors in software (U5.03)
- K7.02 - Identify the decade in which milestones occurred in the development of computing and the Internet (U7.01)
- K7.11 - Describe examples in which computation has or will create new societal phenomena and human capabilities to perceive and act upon our environment (U7.01)
- K7.13 - Describe examples demonstrating that new ways to collaborate and share information are evolving (U7.01)
- K9.04 - Describe career-oriented opportunities to use computational skills to positively affect people's lives (U9.01, U9.03)
- K9.06 - Describe how computing is connected to innovations in other fields (U9.03)

Skills

Students will:

- S1.01 - Create a visual artifact of a computational process, such as an image or screenshot of a user interface or web page (U1.01)
- S4.01 - Communicate a design for a program using natural language, diagrams, and pseudocode (U5.02)
- S5.01 - Design a program by breaking a large plan into smaller modules (U1.03, U5.02, U5.06)
- S5.05 - Use the tools incorporated in an IDE and/or higher-level language to create original programming solutions
- S5.07 - Create a program by incrementally writing and testing modular code (U1.03, U5.03)
- S5.09 - Improve readability, efficiency, or correctness of code from other people (U4.03, U5.03)
- S5.10 - Document a software product using appropriate in-line comments
- S5.13 - Deduce the value of a variable resulting from execution of particular code (U4.03)
- S8.01 - Document a software development process (U8.01)
- S8.02 - Document a software product using high-level documentation (U5.07)
- S8.04 - Effectively manage a project, including planning and time management, team norming and load balancing, file/revision management, and documentation (U8.02)

- S8.05 - Collaborate effectively with others when managing a project (U8.01, U8.02)
- S8.07 - Compare to select from among several tools available for collaboration (U7.01)
- S8.08 - Collaborate when programming (U8.01)
- S8.09 - Collaborate when presenting (U8.01, U8.02)
- S8.12 - Communicate an idea for a product that solves a problem or expresses creativity (U1.01, U8.04, U8.05)
- S8.13 - Communicate the progress on a project, including accomplishments and next steps (U8.05)

Essential Questions

- Q1 - How do computers perform complicated tasks built from simple instructions?
- Q2 - How are variables used in programming?
- Q3 - How do programmers approach a complicated problem?
- Q4 - What role does creativity play in algorithmic programming?
- Q5 - What makes for a good process for collaborative software development?

Day-by-Day Plans

Activity 1.1.1 Principles (2 days) (U7.01, U9.01, U9.03) (K7.02, K9.04, K9.06, K7.11, K7.13)
Activity 1.1.2 Lightbot: Input, Output, State (1 day) (U4.01, U5.06) (K4.01)
Activity 1.1.3 Branching and Iteration (2 days) (U4.02, U4.01) (K5.01) (S8.08)
Activity 1.1.4 Objects and Methods (1 day) (U4.01) (S8.08)
Activity 1.1.5 Variable Roles Part I (1 day) (U4.03, U4.01) (K4.05) (S8.08, S5.13)
Activity 1.1.6 Variable Roles Part II (2 days) *(*U4.03, U4.01) (K4.05) (S8.08, S5.13)
Problem 1.1.7 Scratch Game or Story (6 days)
(U1.01, U1.03, U4.01, U7.05, U8.04, U8.05, U8.01, U5.04, U5.07, U5.03, U5.02, U8.02) (K1.01) (S1.01, S8.08, S8.13, S8.12, S8.09, S8.07, S5.09, S5.07, S5.05, S5.01, S8.04, S8.05, S8.01, S8.02, S4.01, S5.10)

Time: 15 days

Days 1 – 2:

- The teacher will provide the context for the course in terms of the PLTW program, the national effort around College Board's AP CS Principles, and the drastic U.S. shortage of computational specialists.
- Activity 1.1.1 CS Principles frames the course. Use the introduction of the activity and the excerpts of *Blown to Bits* in Steps 6 and 7 to get students thinking about the transformation that is happening in the world. Step 7 can be assigned as homework.
- In Steps 9 and 10, students skim one chapter of *Blown to Bits*. Be explicit that the activity's purpose is to interact with classmates while thinking about the big picture of the course. Let students know that this activity intends to practice a pre-reading strategy (skimming) and presentation skills, but it does not intend to teach the contents of the chapters.
- Step 9 of this activity uses a "jigsaw," with five groups each skimming one of five of the chapters of the book listed in steps 9-10. Groups will report out in step 11. If you have 19 or fewer students, form five groups. If you have 20 or more students, form 10 groups. In

step 11, if you have two groups per chapter, have students listen to both groups before recording their own summary. There is no need to reconcile the two groups' post-skimming speculation.

Day 3:

- The teacher will describe problem-based learning to the students and frame the lesson by presenting Problem 1.1.7. Lead student discussion of the Essential Questions. Guide student attention to the Understandings, Skills, Knowledge, and Key Terms.
- To introduce Activity 1.1.2 Lightbot, the teacher could ask students to recite directions with their partner from somewhere in the school back to the current classroom. “If you direct an agent, why do you have to consider which way they are facing?”
- The teacher will direct pair formation and circulate while students work on Activity 1.1.2, improving pair dynamics and assisting as needed.
- The teacher will present or flip 1.1.2 InputOutputState.pptx.
- Suggest that students complete additional work with Lightbot as homework, and assign the remainder of Activity 1.1.2 as homework. At the beginning of the next session, ask small groups to summarize their journal entry responses to the last conclusion question.

Days 4-5:

- The teacher will introduce Activity 1.1.3 using 1.1.3.A BranchingAndIteration.sb2 available at http://scratch.mit.edu/projects/11520529/
- The teacher will direct pair formation, distribute materials, and circulate while students work, improving pair dynamics and assisting as needed.
- The teacher will present or flip 1.1.3.A HowToProgram.pptx.
- The teacher will present or flip 1.1.3.A BranchingAndIteration.pptx and the associated video.

Day 6:

- The teacher will introduce Activity 1.1.4 Objects and Methods.
- Scratch broadcasts all messages to all sprites, never just to the ones that need to know about the instruction. Multiple sprites can be programmed to respond to the same message. Avoid having multiple sprites respond to a message when demonstrating Scratch to your students. Instead, name the target of an instruction as part of the broadcast message: spriteName.verbishNameForMethodBeingCalled(). Demonstrate object-oriented coding in Lesson 1.1 with the object.method() notation without expecting students to follow suit. In Lesson 1.3 you will introduce the object.method() notation again for the purpose of helping students use built-in methods in *Python*. Devote minimal explanation to object-oriented content and questions at this time. If students ask why the parenthesis, for example, simply say something to the effect that, “The parentheses will come in useful later because *Python* will let you pass arguments – terms we’ll introduce when we get to them in Lesson 1.3. Suffice to say for now that the arguments could be details like the volume or the pitch of the meow.” If students depart from the CSE Scratch convention of spriteName.verb() for naming the broadcast, don’t worry about it. Later activities teach object-oriented conventions (use a verb), syntax (object.method) and concepts, especially Activity 1.4.1 Procedural Abstraction and Activity 1.4.2 Objects and Methods.

- The teacher will direct pair formation, distribute materials, and circulate while students work, improving pair dynamics and assisting as needed.
- Conclude Activity 1.1.4 by having students identify common elements in software that require one component to call a method of another component.

Day 7:

- The teacher will address a common misconception by telling students that a variable in computer programming represents particular contents of the computer's memory. The teacher will ask students to brainstorm how that is different from the meaning of the term "variable" in mathematics.
- The teacher will present or flip the first half of 1.1.5 RolesOfVariables.pptx (covering the first four roles), perhaps using the Scratch demonstration programs provided for the Activity.
- The teacher will introduce Activity 1.1.5 Variable Roles I and direct pair formation, distribute materials, and circulate while students work, improving pair dynamics and assisting as needed.
- Conclude Activity 1.1.5 by revisiting the question of how the term "variable" is different in computer science than in mathematics.

Days 8 - 9:

- Collect student work on Activity 1.1.5 and discuss selected questions as necessary.
- The teacher will present the second half of 1.1.5 Roles of Variables (covering the last four roles), perhaps using the Scratch demonstration programs provided for the Activity.
- The teacher will direct pair formation, distribute materials, and circulate while students work on Activity 1.1.6 Variable Roles II, improving pair dynamics and assisting as needed.
- Approximately 10% of teachers in the 2013-2014 pilot of Computer Science and Software Engineering had students work independently outside of class on *Python* using http://www.codecademy.com/python beginning in the last week of Lesson 1.1 or the first week of Lesson 1.2.

Days 10 - 15:

- The teacher will present 1.1.7 SoftwareDesignProcess.pptx and guide students as they reference http://scrumreferencecard.com/ScrumReferenceCard.pdf.
- The teacher will lead discussion about what makes for an effective team dynamic.
- The teacher will direct pair formation and circulate while students brainstorm, identify a product, specify a task list, and proceed to work on Problem 1.1.7 Scratch Game or Story. The teacher will improve pair dynamics and assist as needed.
- The teacher will lead discussion about effective presentation skills using the rubric for Problem 1.1.7 as a guide, perhaps using a one-way flag exercise to demonstrate eye contact in which each audience member raises her hand and keeps it raised (like a one-way flag) if the presenter makes eye contact with her.
- The teacher will provide opportunity for students to present their products to other students and collect them for evaluation.

Unit 1.2 App Design

The goal of this unit is for students to build an Android app of their own design. Students first build their skills by analyzing and creating code, especially by thinking about the roles of variables. The unit begins with an introduction to binary representation of numbers, letters, colors, images, and any other digital data. Students are introduced to data abstraction, which is the idea that programs can handle complicated data like images without the programmer having to worry about low-level details like zeroes and ones. Students work with and make minor modifications to two App Inventor programs and build their ability to analyze a complex program. They conclude the lesson by designing and creating an Android app, practicing pair programming, and applying the Agile software design process.

Understandings

- U1.01 - Computing fosters creative expression, sometimes resulting in artifacts. (K1.03, K1.01) (S1.01, S1.04, S8.12)
- U1.03 - Programming is a creative endeavor. (S5.07, S5.03, S5.01)
- U2.01 - Binary sequences represent digital data. (K2.02, K2.03, K2.01, K2.16) (S2.06, S2.07)
- U2.06 - The solution to one problem can be applied to another seemingly unrelated problem by identifying and reusing a pattern. (S5.03)
- U2.08 - Physical systems, like sound or biological molecules, have both digital and analog characteristics.
- U4.01 - Programs implement algorithms to solve problems. (K4.05, K4.06)
- U4.03 - Empirical analysis of algorithms requires a systematic approach. (S5.09, S5.02, S5.13)
- U5.01 - Creating solutions with computation requires exploring the tools available, selecting an appropriate tool, and gaining expertise with the tool. (S5.05, S5.03)
- U5.02 - Solutions in a programming language are created by breaking a problem apart into component problems. (K4.06) (S5.01, S4.01, S5.11)
- U5.03 - Creating solutions with computation requires a persistent, iterative problem-solving approach. (K5.01) (S5.09, S5.07)
- U5.04 - Programming requires an understanding of mathematical operations and data abstractions. (K4.05) (S2.07)
- U5.06 - Functions with arguments make code modular and reusable. (S5.01, S5.11)
- U7.05 - New opportunities for human creativity and innovation exist because of networked, mobile, and embedded computing. (K1.03, K1.01)
- U8.01 - Computing artifacts and programs can be higher quality as a result of collaboration. (S8.08, S8.09, S8.05, S8.01)
- U8.02 - Working in a team requires effective communication, clear responsibilities, and attention to interpersonal relationships. (S8.09, S8.04, S8.05)
- U8.04 - Creative ideas and technical solutions must be communicated in a clear and concise manner. (S8.12)
- U8.05 - How people present themselves affects how their work is received. (S8.13, S8.12)

Knowledge

Students will:

- K1.01 - Describe the role of creativity in designing an attractive, functional, and accessible graphical user interface. (U1.01, U7.05)
- K1.03 - Describe the role of creativity in designing a mobile application to solve a problem. (U1.01, U7.05)
- K2.01 - Describe the ways in which various types of digital data can be represented in binary. (U2.01)
- K2.02 - Distinguish continuous and discrete phenomena and identify digital and analog data. (U2.01)
- K2.03 - Describe layers of abstraction that help people represent and use data. (U2.01)
- K2.04 - Identify layers of abstraction used in programming languages and describe advantages and disadvantages inherent in working at a high level of abstraction.
- K2.16 - Describe the implications of the limited precision of digital information in applications. (U2.01)
- K4.05 - Recognize common patterns employing variables, including value accumulation, list aggregation, and iteration across the elements of a collection. (U4.01, U5.04)
- K4.06 - Recognize events and event-handlers implied by a user interface. (U4.01, U5.02)
- K5.01 - Describe ways to identify the existence and location of errors in software. (U5.03)
- K9.04 - Describe career-oriented opportunities to use computational skills to positively affect people's lives.
- K9.06 - Describe how computing is connected to innovations in other fields.

Skills

Students will:

- S1.01 - Create a visual artifact of a computational process, such as an image or screenshot of a user interface or web page. (U1.01)
- S1.04 - Create a graphical user interface. (U1.01)
- S2.06 - Consider implications of converting data from one representation to another, for example noise or compression. (U2.01)
- S2.07 - Convert a given quantity among bases 2, 10, and 16. (U2.01, U5.04)
- S4.01 - Communicate a design for a program using natural language, diagrams, and pseudocode. (U5.02)
- S5.01 - Design a program by breaking a large plan into smaller modules. (U1.03, U5.02, U5.06)
- S5.02 - Analyze and test code from other people. (U4.03)
- S5.03 - Extend, or apply to new purpose, code from other people. (U1.03, U2.06, U5.01)
- S5.05 - Use the tools incorporated in an IDE and/or higher-level language to create original programming solutions. (U5.01)
- S5.07 - Create a program by incrementally writing and testing modular code. (U1.03, U5.03)
- S5.09 - Improve readability, efficiency, or correctness of code from other people. (U4.03, U5.03)
- S5.10 - Document a software product using appropriate in-line comments.

- S5.11 - Encapsulate a set of related statements in a function or procedure. (U5.02, U5.06)
- S5.13 - Deduce the value of a variable resulting from execution of particular code. (U4.03)
- S8.01 - Document a software development process. (U8.01)
- S8.02 - Document a software product using high-level documentation.
- S8.04 - Effectively manage a project, including planning and time management, team norming and load balancing, file/revision management, and documentation. (U8.02)
- S8.05 - Collaborate effectively with others when managing a project. (U8.01, U8.02)
- S8.07 - Compare to select from among several tools available for collaboration.
- S8.08 - Collaborate when programming. (U8.01)
- S8.09 - Collaborate when presenting. (U8.01, U8.02)
- S8.12 - Communicate an idea for a product that solves a problem or expresses creativity. (U1.01, U8.04, U8.05)
- S8.13 - Communicate the progress on a project, including accomplishments and next steps. (U8.05)

Essential Questions

- ➢ Q1 - What do programming languages and development environments have in common?
- ➢ Q2 - What can be represented by binary data?
- ➢ Q3 - What makes for a good software development process?
- ➢ Q4 - How can a program be analyzed, understood, and modified?

Day-by-Day Plans

Activity 1.2.1	Bits and Bytes (1 day) (U5.04, U2.01, U2.08) (K2.02, K2.03, K2.01, K9.06, K2.16) (S2.06, S2.07)
Activity 1.2.2	Introducing App Inventor (2 days) (U4.01)
Activity 1.2.3	Creating Mobile Apps (2 days) (U4.01) (K4.06)
Activity 1.2.4	Analyzing a Program (2 days) (U4.03, U4.01, U5.06, U5.02) (K5.01, K4.05) (S8.08, S5.02, S4.01, S5.13)
Project 1.2.5	Modifying a Program (2 days) (U4.03, U4.01, U2.06) (K9.04) (S8.08, S5.09, S5.03)
Problem 1.2.6	Designing an App (6 days) (U1.01, U1.03, U4.01, U7.05, U8.04, U8.05, U8.01, U5.01, U5.03, U5.02, U8.02) (K1.03, K1.01, K2.04) (S1.01, S1.04, S8.08, S8.13, S8.12, S8.09, S8.07, S5.07, S5.05, S5.01, S8.04, S8.05, S8.01, S8.02, S4.01, S5.10, S5.11)

Time: 15 days

Day 1:

- The teacher will frame the lesson by presenting Problem 1.2.6 and by leading student discussion of the Essential Questions. Guide student attention to the Understandings, Skills, Knowledge, and Key Terms.

- The teacher will present the binary number system using 1.2.1 BitsAndBytes.pptx and introduce Activity 1.2.1.
- The teacher will direct pair formation, distribute materials, and circulate while students work, improving pair dynamics and assisting as needed.
- The teacher will assign the remainder of Activity 1.2.1 as homework. Students can practice using the 1.2.1 Canvas Quiz Binary2Decimal until they demonstrate mastery.
- Building on questions 13 and 14, a student who is interested in some of the anecdotes surrounding the use of digital data and reliance on computers will find interesting reading at catless.ncl.ac.uk/Risks/index.27.html . This newsletter is updated weekly with interesting bugs and news of hazards from the software industry.
- Question 16 provides a tie-in for *Blown to Bits*, Chapter 6. http://www.bitsbook.com/wp-content/uploads/2008/12/chapter6.pdf. Consider using an excerpt.

Days 2-3:

- Discuss selected questions from Activity 1.2.1 Bits and Bytes and collect student work for evaluation.
- To introduce Activity 1.2.2 Introducing App Inventor, the teacher will demonstrate the starter code for the color_chooser app (file name color_chooser_stage1). The teacher will also demonstrate the finished form of the color_chooser app (file name color_chooser_AK) from the Teacher Resources file folder. These demonstrations could be performed by having students view the videos linked from within Activity 1.2.2, by having students experiment with the apps on an Android device, or by using a document camera to project the screen from an Android device with the apps.
- The teacher will provide a brief tour of the App Inventor environment.
- The teacher will direct pair formation, distribute materials, and circulate while students work, improving pair dynamics and assisting as needed.
- Conclude Activity 1.2.2 by having students share their responses to the conclusion questions and reconsidering the lesson's essential questions.
- Make sure that at the end of each session, students log out of their Google account on the computer and (if applicable) delete their Google account from the Android device.

Days 4-5:

- Demonstrate Activity 1.2.3 by building the app in front of students. To ensure the technology is working and that students can navigate the tool from beginning to end, check that students have created their own app and have transferred it to a device.
- Demonstrate portions of the problems suggested. How thoroughly you should demonstrate building the code in blocks will depend on your students’ desire to follow along vs. figure things out. You could lead some groups more than others if students are eager to experiment or desirous of guidance.

Days 6-7:

- To introduce Activity 1.2.4 Analyzing a Program, the teacher will
    - ➢ activate student knowledge about the roles of variables by asking how one or more of the roles from Lesson 1.1 might be used in an app.

    - ➢ introduce the concept of a procedure by giving an example of a procedure like "get ready for school" that includes many steps but that can be referred to with the single name "get ready for school." Ask students to name a procedure they commonly refer to.
- The teacher will direct pair formation, distribute materials, and circulate while students work, improving pair dynamics and assisting as needed.
- In steps 9-15, students analyze the code one stage at a time, and the complicated program is presented in four earlier stages that incorporate only a portion of the program's features. These earlier stages are to be viewed in the Blocks editor. Students might be successful without viewing the simpler stages of the program on the Android device. If you choose to have students view the simpler stage on the Android devices, consider first letting all students install the full program and explore it. Consider then dividing the class up into thirds for downloading stages 1, 2, and 3 to minimize class time spent downloading and installing all stages. Alternatively, pre-install all stages on all Android devices.
- Step 15 is an optional extension because the amount of code introduced in this stage may be overwhelming for less resilient students. Advanced students should absolutely do this as a way to reinforce best-so-far and one-way flag variable roles.
- Conclude Activity 1.2.4 by leading student discussion comparing the roles of variables and development features in the Scratch™ and App Inventor environments.

Days 8-9:

- To introduce Project 1.2.5 Modifying a Program, the teacher will ask students what they think would make the program from the previous activity more interesting. Ask students if there are features they have wished for in computer applications they use. Ask students how they think development teams prioritize which features to add to software and how they accomplish those changes.
- Because of the large amount of time consumed by the download->email->download->install->run cycle, Part II may become cumbersome if you are not using a method for live connection to the Android device. If you decide to skip Part II for time management reasons, make sure to inform students of the importance of error detection and correction. Let them know that you will teach this skill as errors arise in the classroom.

Days 10-15:

- To introduce Problem 1.2.6, the teacher will review the Scrum process and strategies for working effectively in a group.
- The teacher will direct pair formation, distribute materials, and circulate while students brainstorm, identify a product, specify a task list, and proceed to work on Problem 1.2.6. The teacher will improve pair dynamics and assist as needed.
- The teacher will provide opportunity for students to present their products to other students and collect them for evaluation.
- Conclude Problem 1.2.6 by providing time for students to present their work to another group.

Unit 1.3 Algorithms in *Python*

The goal of this unit is for students to understand all information as bits and to transfer their understanding of algorithms to a new language. Students examine various games such as mastermind, hangman, and the Prisoner's Dilemma and have the opportunity to be creative with their own visual work. Students are introduced to functional, imperative, and declarative programming paradigms with the *Python*® programming language, again learning to use variables in the most common roles. Variable types, arrays, and array manipulations are introduced. Students learn how data is digitally represented, making macroscopic magnetic storage in an unplugged activity and manipulating image files using both application programming interfaces (APIs) and direct manipulation of the data. Students read, discuss, and write about intellectual property and privacy issues that stem from our ability to reproduce and automate analysis of multimedia data. Students conclude the unit with two activities: a collaborative project to manipulate data with Python, creating an image artifact, and the creation of a software program that can be used for a round-robin tournament of a strategy game which applies the concepts of game theory.

Understandings

- U1.03 - Programming is a creative endeavor. (S5.01)
- U2.01 - Binary sequences represent digital data. (K2.03)
- U2.02 - Computing relies on layers of abstraction in software. (S5.11, S5.12)
- U2.05 - Solutions to complex problems can be encapsulated in reusable components. (S5.11, S5.12)
- U4.01 - Programs implement algorithms to solve problems. (K5.02, K4.05, K4.01) (S5.12)
- U4.02 - Algorithms can be analyzed for efficiency, and appropriate algorithms can be selected based upon efficiency. (S4.02)
- U4.03 - Empirical analysis of algorithms requires a systematic approach. (S5.09, S5.08, S5.02, S5.13)
- U5.02 - Solutions in a programming language are created by breaking a problem apart into component problems. (S5.01, S4.01, S5.11)
- U5.03 - Creating solutions with computation requires a persistent, iterative problem-solving approach. (K5.01) (S5.09)
- U5.04 - Programming requires an understanding of mathematical operations and data abstractions. (K5.02, K4.05) (S5.08)
- U5.06 - Functions with arguments make code modular and reusable. (K5.02) (S5.01, S5.11, S5.12)
- U5.08 - Programmers must prioritize making their code well documented and readable for it to be maintained. (S5.09, S5.06, S5.10)
- U7.05 - New opportunities for human creativity and innovation exist because of networked, mobile, and embedded computing.
- U8.01 - Computing artifacts and programs can be higher quality as a result of collaboration. (S8.08)
- U8.02 - Working in a team requires effective communication, clear responsibilities, and attention to interpersonal relationships.
- U8.03 - Collaboration allows communities to create software that can impact people's lives. (K8.01) (S8.08, S8.03)

- U9.02 - Parallel computing is a quickly evolving field relevant to hardware, software, and users.

Knowledge

Students will:

- K2.01 - Describe the ways in which various types of digital data can be represented in binary. (U2.01)
- K2.03 - Describe layers of abstraction that help people represent and use data. (U2.01)
- K4.01 - Describe a computer as responding to input in a deterministic manner that depends only on input and on the computer's state. (U4.01)
- K4.05 - Recognize common patterns employing variables, including value accumulation, list aggregation, and iteration across the elements of a collection. (U4.01, U5.04)
- K5.01 - Describe ways to identify the existence and location of errors in software. (U5.03)
- K5.02 - Describe the role that functions play in developing software. (U4.01, U5.04, U5.06)
- K7.04 - Articulate a range of positions on the question of ownership of bits.
- K8.01 - Describe a version control system. (U8.03)
- K9.02 - Describe the impact that computing has had in the social sciences, geography, and civics.

Skills

Students will:

- S4.01 - Communicate a design for a program using natural language, diagrams, and pseudocode. (U5.02)
- S4.02 - Evaluate a program for efficiency. (U4.02)
- S5.01 - Design a program by breaking a large plan into smaller modules. (U1.03, U5.02, U5.06)
- S5.02 - Analyze and test code from other people. (U4.03)
- S5.05 - Use the tools incorporated in an IDE and/or higher-level language to create original programming solutions.
- S5.06 - Evaluate programs written by others for readability. (U5.08)
- S5.08 - Identify appropriate boundary conditions for testing a program. (U4.03, U5.04)
- S5.09 - Improve readability, efficiency, or correctness of code from other people. (U4.03, U5.03, U5.08)
- S5.10 - Document a software product using appropriate in-line comments. (U5.08)
- S5.11 - Encapsulate a set of related statements in a function or procedure. (U2.02, U2.05, U5.02, U5.06)
- S5.12 - Create a function to perform a particular calculation from the function's arguments. (U2.02, U2.05, U4.01, U5.06)
- S5.13 - Deduce the value of a variable resulting from execution of particular code. (U4.03)
- S8.03 - Use a version control system. (U8.03)
- S8.07 - Compare to select from among several tools available for collaboration.
- S8.08 - Collaborate when programming. (U8.01, U8.03)

Essential Questions

- Q1 – What can be represented by binary data?
- Q2 – How does abstraction make the software development process easier?
- Q3 – What are the practices that lead to effective collaboration?
- Q4 – What role does creativity play in algorithmic programming?

Day-by-Day Plans

Activity 1.3.1	Programs are Data (1 day) (U4.01, U9.02, U2.01, U2.02) (K2.01, K2.03, K4.01)
Activity 1.3.2	*Python* Variables and Functions (2 days) (U4.01, U5.04, U5.06, U5.02) (K5.02, K4.05) (S8.08, S5.11, S5.12)
Activity 1.3.3	Branching and Output (2 days) (U4.02, U4.01, U5.04) (K5.02, K5.01) (S8.08, S5.08, S4.02, S5.12)
Activity 1.3.4	Nested Branching and Input (2 days) (U4.03, U4.01, U5.04) (K5.02, K5.01) (S8.08, S5.08, S5.12)
Activity 1.3.5	Strings (2 days) (U4.03, U4.01, U5.04, U5.08) (K5.02, K5.01, K4.05) (S8.08, S5.08, S5.06, S5.01, S5.11, S5.12, S5.13)
Activity 1.3.6	Tuples and Lists (3 days) (U4.03, U4.01, U5.04, U5.08) (K5.02, K5.01, K4.05) (S8.08, S5.08, S5.06, S5.01, S5.11, S5.12, S5.13)
Activity 1.3.7	For Loops (3 days) (U4.03, U4.01, U5.04, U5.08) (K5.02, K5.01, K4.05) (S8.08, S5.08, S5.06, S5.01, S5.11, S5.12, S5.13)
Activity 1.3.8	While Loops (2 days) (U4.03, U4.01, U5.04, U5.08) (K5.02, K5.01, K4.05) (S8.08, S5.08, S5.06, S5.01, S5.11, S5.12, S5.13)
Project 1.3.9	Tools for Collaboration (3 days) (U1.03, U4.03, U4.01, U7.05, U8.03, U8.01, U5.03, U2.05, U8.02) (K8.01, K7.04, K9.02) (S8.08, S8.07, S5.09, S5.05, S5.02, S8.03, S4.01, S5.10)

Time: 20 days

Day 1:

- Frame the lesson by presenting Problem 1.3.9 and by leading student discussion of the Essential Questions. Guide student attention to the Understandings, Skills, Knowledge, and Key Terms.
- Present 1.3 Abstraction.pptx
- Review binary numbers with a focus on abstraction; use Activity 1.2.1 as a resource.
- Introduce Activity 1.3.1 by summarizing information provided in the Introduction and Background sections of the activity.
- Circulate while students work on Activity 1.3.1 steps 1 and 2 during class.
- Assign steps 3 and 4 and the Conclusion questions as homework.

Days 2-3:

- Discuss selected questions from Activity 1.3.1.
- Present 1.3.2 Roles of Variables, activating prior knowledge from AI and Scratch.
- Present 1.3.2 *Python* Basics or use the video found at http://www.youtube.com/watch?v=RBE5ZU1KiH8
- Demonstrate use of Canopy for students. Highlight the distinction between iPython interactive session and the code editor. Include an explanation and demonstration of the process for logging and retrieving log files in Canopy.
- Direct pair formation, distribute materials, and circulate while students work through Activity 1.3.2, improving pair dynamics and assisting as needed.
- Assign conclusion questions as homework.

Days 4-7:

- Discuss selected questions from Activity 1.3.2.
- Think-pair-share about input/output and responsiveness to input within an app that each student has used (per Introduction in Activity 1.3.3)
    - ➢ Introduce Flow Charts with http://boingboing.net/images/baconflowschart.jpg or another humorous flowchart of your own choosing.
    - ➢ Summarize how the responsiveness of an app or program can be modeled by a flow chart.
- Direct pair formation, distribute materials, and circulate while students work through Activity 1.3.3 and Activity 1.3.4, improving pair dynamics and assisting as needed.
- Conclude Activities 1.3.3 and 1.3.4 by having students share their responses to the conclusion questions and reconsidering the essential questions.

Days 8-17:

- Introduce tuples, as well as looping through the elements of iterables. Create a short program in front of students that creates a tuple with several kinds of data structures, using a for loop to print the contents of the tuple. See SampleTupleProgram.py in the 1.3.6 teacherSourceFiles for ideas. Discuss what collections like a tuple might be used for in various programs.
- Direct pair formation, distribute materials, and circulate while students work through Activity 1.3.5-1.3.8, improving pair dynamics and assisting as needed.
- Conclude Activities 1.3.5-1.3.8 by having students share their responses to the conclusion questions.

Days 18-20:

- Introduce the Prisoner's Dilemma by modeling it with two volunteers from the audience. Make up a crime.
  - Explain the penalty/reward table shown in Activity 1.3.9.
  - Have each volunteer secretly record a decision to collude or betray.
  - Record the score for each volunteer for the round.
  - Repeat for 3 or 4 iterations.
  - Compare the final score to the best score each individual could have attained and to the best combined score the pair could have attained.
- Model how to retrieve source code through Git. Remind students that they will need one clone to test on and one to merge with the repository.
- Direct pair formation, distribute materials, and circulate while students work through Activity 1.3.9, improving pair dynamics and assisting as needed.
- Conclude Activity 1.3.9 by having students share their responses to the conclusion questions.

Unit 1.4 Images and Object-Oriented Libraries

The goal of this unit is for students to become independent learners of a programming language, able to reference documentation to use object-oriented libraries commonly available to their language of study. The unit begins with an unplugged activity to teach object-oriented concepts. Students build additional strength with *Python* algorithms, manipulating image files by modifying pixel data and using code libraries to work at higher levels of abstraction. As part of that work, they learn to use various sources of documentation, including application programming interfaces (APIs). Students read, discuss, and debate intellectual property issues associated with digital data. In the culminating problem of the lesson, they collaborate to create an image processing function that highlights the power of automation.

Understandings

- U1.01 - Computing fosters creative expression, sometimes resulting in artifacts (S1.01, S8.12)
- U1.02 - Computational artifacts can be evaluated (S1.05)
- U1.03 - Programming is a creative endeavor (S5.07, S5.03, S5.01)
- U2.01 - Binary sequences represent digital data (K2.03, K2.01, K7.08, K2.16) (S2.06)
- U2.02 - Computing relies on layers of abstraction in software (K2.04, K2.16) (S5.11, S5.12)
- U2.04 - Abstraction allows for simple utilization of other people's code (S5.04, S5.03)
- U2.05 - Solutions to complex problems can be encapsulated in reusable components (K5.03) (S5.11, S5.12)
- U4.01 - Programs implement algorithms to solve problems (K5.02, K4.05, K4.01) (S5.12)
- U4.02 - Algorithms can be analyzed for efficiency, and appropriate algorithms can be selected based upon efficiency (S4.02)
- U4.03 - Empirical analysis of algorithms requires a systematic approach (S5.09, S5.08, S5.02, S5.13)

- U5.01 - Creating solutions with computation requires exploring the tools available, selecting an appropriate tool, and gaining expertise with the tool (K5.03, K5.02) (S5.05, S5.04, S5.03, S8.11)
- U5.02 - Solutions in a programming language are created by breaking a problem apart into component problems (S5.01, S4.01, S5.11)
- U5.03 - Creating solutions with computation requires a persistent, iterative problem-solving approach (K5.01) (S5.09, S5.07)
- U5.04 - Programming requires an understanding of mathematical operations and data abstractions (K5.02, K4.05) (S5.08)
- U5.06 - Functions with arguments make code modular and reusable (K5.02) (S5.01, S5.11, S5.12)
- U5.07 - Programmers create high-level documentation to communicate the purpose and function of their code (K5.03)
- U5.08 - Programmers must prioritize making their code well-documented and readable for it to be maintained (S5.09, S5.06, S5.10)
- U7.01 - Mobile and networked computing have transformed commerce, social interactions, news sourcing and dissemination, and culture (K8.01, K7.04) (S7.02, S8.07)
- U7.05 - New opportunities for human creativity and innovation exist because of networked, mobile, and embedded computing
- U8.01 - Computing artifacts and programs can be higher quality as a result of collaboration (S8.08, S8.09, S8.05, S8.01)
- U8.02 - Working in a team requires effective communication, clear responsibilities, and attention to interpersonal relationships (S8.09, S8.04, S8.05)
- U8.03 - Collaboration allows communities to create software that can impact people's lives (K8.01) (S8.08, S8.03)
- U8.04 - Creative ideas and technical solutions must be communicated in a clear and concise manner (S8.12)
- U8.05 - How people present themselves affects how their work is received (S8.10, S8.13, S8.12)
- U9.02 - Parallel computing is a quickly evolving field relevant to hardware, software, and users

Knowledge

Students will:

- K2.01 - Describe the ways in which various types of digital data can be represented in binary (U2.01)
- K2.03 - Describe layers of abstraction that help people represent and use data (U2.01)
- K2.04 - Identify layers of abstraction used in programming languages and describe advantages and disadvantages inherent in working at a high level of abstraction (U2.02)
- K2.16 - Describe the implications of the limited precision of digital information in applications (U2.01, U2.02)
- K4.01 - Describe a computer as responding to input in a deterministic manner that depends only on input and on the computer's state (U4.01)
- K4.05 - Recognize common patterns employing variables, including value accumulation, list aggregation, and iteration across the elements of a collection (U4.01, U5.04)

- K5.01 - Describe ways to identify the existence and location of errors in software (U5.03)
- K5.02 - Describe the role that functions play in developing software (U4.01, U5.01, U5.04, U5.06)
- K5.03 - Distinguish among a variety of educational and reference resources related to code libraries (U2.05, U5.01, U5.07)
- K7.04 - Articulate a range of positions on the question of ownership of bits (U7.01)
- K7.05 - Describe what metadata contain and how they can be used
- K7.08 - Discuss societal implications of the persistence and the ease of copying digital information (U2.01)
- K8.01 - Describe a version control system (U7.01, U8.03)
- K9.02 - Describe the impact that computing has had in the social sciences, geography, and civics

Skills

Students will:

- S1.01 - Create a visual artifact of a computational process, such as an image or screenshot of a user interface or web page (U1.01)
- S1.05 - Analyze a computational artifact for usability with a specific audience in mind (U1.02)
- S2.06 - Consider implications of converting data from one representation to another, for example noise or compression (U2.01)
- S4.01 - Communicate a design for a program using natural language, diagrams, and pseudocode (U5.02)
- S4.02 - Evaluate a program for efficiency (U4.02)
- S5.01 - Design a program by breaking a large plan into smaller modules (U1.03, U5.02, U5.06)
- S5.02 - Analyze and test code from other people (U4.03)
- S5.03 - Extend, or apply to new purpose, code from other people (U1.03, U2.04, U5.01)
- S5.04 - Explore and use documentation and public information to extend the student's own knowledge of a programming language or to achieve a computational approach to solve a problem (U2.04, U5.01)
- S5.05 - Use the tools incorporated in an IDE and/or higher-level language to create original programming solutions (U5.01)
- S5.06 - Evaluate programs written by others for readability (U5.08)
- S5.07 - Create a program by incrementally writing and testing modular code (U1.03, U5.03)
- S5.08 - Identify appropriate boundary conditions for testing a program (U4.03, U5.04)
- S5.09 - Improve readability, efficiency, or correctness of code from other people (U4.03, U5.03, U5.08)
- S5.10 - Document a software product using appropriate in-line comments (U5.08)
- S5.11 - Encapsulate a set of related statements in a function or procedure (U2.02, U2.05, U5.02, U5.06)
- S5.12 - Create a function to perform a particular calculation from the function's arguments (U2.02, U2.05, U4.01, U5.06)

- S5.13 - Deduce the value of a variable resulting from execution of particular code (U4.03)
- S7.02 - Work with a group to agree on a policy or protocol (U7.01)
- S8.01 - Document a software development process (U8.01)
- S8.03 - Use a version control system (U8.03)
- S8.04 - Effectively manage a project, including planning and time management, team norming and load balancing, file/revision management, and documentation (U8.02)
- S8.05 - Collaborate effectively with others when managing a project (U8.01, U8.02)
- S8.07 - Compare to select from among several tools available for collaboration (U7.01)
- S8.08 - Collaborate when programming (U8.01, U8.03)
- S8.09 - Collaborate when presenting (U8.01, U8.02)
- S8.10 - Present original research and research from literature (U8.05)
- S8.11 - Distinguish sources of information and comment on the information's reliability and intended audience (U5.01)
- S8.12 - Communicate an idea for a product that solves a problem or expresses creativity (U1.01, U8.04, U8.05)
- S8.13 - Communicate the progress on a project, including accomplishments and next steps (U8.05)

Essential Questions

- ➢ Q1 – What can be represented by binary data?
- ➢ Q2 – How does abstraction make the software development process easier?
- ➢ Q3 – What are the practices that lead to effective collaboration?
- ➢ Q4 – What role does creativity play in algorithmic programming?

Day-by-Day Plans

Activity 1.4.1	Procedural Abstraction (1 day)
Activity 1.4.2	Objects and Methods (2 days)
Activity 1.4.3	Images and Arrays (2 days)
Activity 1.4.4	*Python* Imaging Library API (2 days)
Project 1.4.5	Image Algorithms (3 days)
Activity 1.4.6	Digital Property and Forensics (2 days)
Problem 1.4.7	Image Artist (5 days)

Time: 17 days

Day 1:

- The teacher will frame the lesson by presenting Problem 1.4.7 and by leading student discussion of the Essential Questions. Guide student attention to the Understandings, Skills, Knowledge, and Key Terms.
- Present 1.4.1 Abstraction.pptx.
- Activity 1.4.1 uses ping pong balls; the code RealityArtist does not actually execute on a computer; you and students pretend to be the computer. Ping-pong balls are instantiated when you provide a ping-pong ball to a student. A method on a particular ping-pong ball is executed when a student draws on the ping-pong ball.

- The source code in example_class.py can be executed on a computer. Commands to call the __init__ constructor and the grow() method are provided in comments in that source code and can be performed as a demonstration with students.

Days 2-5:

- Assign Step 1 from Activity 1.4.6. Students will read Chapter 6 of “Blown To Bits” as homework.
- Introduce image manipulation by having the class brainstorm about when and how computers use images. Highlight and promote diversity of ideas here.
- Direct pair formation, distribute materials, and circulate while students work through Activities 1.4.2-1.4.3, improving pair dynamics and assisting one another as needed.
- Conclude Activities 1.4.2-1.4.3 by having students share their responses to the conclusion questions.
- The face recognition program used in Activity 3.2.2 can be used in conjunction with the last step in Activity 1.4.2. Demonstrate the face recognition software and ask, "How does this program decide where the eyes are?"

Days 6-10:

- Introduce Activities 1.4.4-1.4.5 by showing students the package manager from Canopy’s opening screen. Ask students why someone would use these libraries instead of writing the code themselves.
- Review 1.4 ReferenceCardForPyplotAndPIL with students, emphasizing the concepts of methods and attributes.
- Direct pair formation, distribute materials, and circulate while students work through Activities 1.4.4-1.4.5, improving pair dynamics and assisting as needed.
- Step 10b in Activity 1.4.5 is an open-ended problem:
    - ➢ "Create a function alter_all_images() that makes a new version of all pictures in a directory, with the modification being of your own design."
    - ➢ To condense the number of days spent at the end of this lesson, student work for this step can satisfy Problem 1.4.7 by creating a client-developer dynamic (perhaps using one of the clients from Problem 1.4.7 who all want "an automated process to apply to images." If using Step 10b as the problem for Lesson 1.4, emphasize the documentation, the presentation, and the reflection on the development and team processes.
- Conclude Activities 1.4.4-1.4.5 by having students share their responses to the conclusion questions.
- As an optional extension to Activity 1.4.5, show students how to automatically generate API documentation from the three function's docstrings. Use the opportunity to critique the docstrings. Give the bigger picture regarding the levels and details belonging in various types of documentation (docstrings, commented lines describing sections of code, inline comments, and documentation that is not embedded in the code).

Day 11:

- Discuss the reading from *Blown to Bits* Chapter 6.

- Introduce the debate with the resolution, asking students to share their initial opinions about the resolution.
- Facilitate formation of teams and organization of team responsibilities as teams prepare for a debate described in step 4 of Activity 1.4.6.
- Assign continuation of step 4 as homework.

Day 12:

- To introduce Problem 1.4.7, review the Scrum process with students using the Scrum reference card and strategies for working effectively in a group.
- The teacher will direct pair formation, distribute materials, and circulate while students brainstorm, identify a product, specify a task list, and proceed to work on Project 1.4.7. The teacher will improve pair dynamics and assist as needed.
- Remind students to prepare for the debate on the next class day.

Day 13:

- Run the debate in Activity 1.4.6, possibly using the format suggested in step 4.
- Conclude Activity 1.4.6 by having students share their responses to the conclusion questions.

Day 14-17:

- Continue work on Problem 1.4.7 where you left off on day 12.
- Provide opportunity for students to present their products to other students and collect them for evaluation.

Lesson 1.5 GUIs in *Python*

In this unit, students will build on their ability to use object-oriented programming and an API to create a graphical user interface (GUI) that is designed for a particular audience, and which considers user accessibility. The unit begins with an unplugged activity that generalizes the user interface topic of this lesson to the field of human-computer interaction. Students practice using an application programming interface (API) to learn methods to affect an object’s state. Students work with two APIs: the Tkinter Canvas for drawing and animation, and then the Tkinter toolbox of GUI widgets. Students are provided code for a simple GUI that implements a model-view-controller (MVC) pattern. Students will modify the elements of that pattern to suit their own needs. The lesson concludes with a problem in which students create a model-view-controller GUI using Scratch or *Python*. Strategies for documentation are reinforced, and Agile development is emphasized in the concluding problem.

Understandings

- U1.01 - Computing fosters creative expression, sometimes resulting in artifacts (K1.01) (S1.01, S1.04, S8.12)
- U1.02 - Computational artifacts can be evaluated (S1.05)
- U1.03 - Programming is a creative endeavor (S5.07, S5.03, S5.01)
- U2.02 - Computing relies on layers of abstraction in software (K2.04) (S5.11)
- U2.04 - Abstraction allows for simple utilization of other people's code (S5.04, S5.03)

- U2.05 - Solutions to complex problems can be encapsulated in reusable components (K5.03, K2.10) (S5.11)
- U2.06 - The solution to one problem can be applied to another seemingly unrelated problem by identifying and reusing a pattern (K2.10, K4.02) (S5.03)
- U4.01 - Programs implement algorithms to solve problems (K5.02, K4.05, K4.06, K4.02)
- U4.03 - Empirical analysis of algorithms requires a systematic approach (S5.09, S5.08)
- U4.04 - A given algorithmic problem with standard solutions can be applied in diverse contexts (K4.02) (S5.03)
- U5.01 - Creating solutions with computation requires exploring the tools available, selecting an appropriate tool, and gaining expertise with the tool (K5.03, K5.02) (S5.05, S5.04, S5.03, S8.11)
- U5.02 - Solutions in a programming language are created by breaking a problem apart into component problems (K4.06) (S5.01, S4.01, S5.11)
- U5.03 - Creating solutions with computation requires a persistent, iterative problem-solving approach (K5.01) (S5.09, S5.07)
- U5.04 - Programming requires an understanding of mathematical operations and data abstractions (K5.02, K4.05) (S5.08)
- U5.05 - The user interface of a piece of software can greatly affect how it is used (S1.05, S1.04)
- U5.06 - Functions with arguments make code modular and reusable (K5.02) (S5.01, S5.11)
- U5.07 - Programmers create high-level documentation to communicate the purpose and function of their code (K5.03) (S8.02)
- U5.08 - Programmers must prioritize making their code well-documented and readable for it to be maintained (S5.09, S5.06, S5.10)
- U7.03 - Assistive technologies using hardware and software can extend human capabilities (K7.11)
- U7.04 - Making information accessible to all people requires attention from a variety of stakeholders (S1.05)
- U8.01 - Computing artifacts and programs can be higher quality as a result of collaboration (S8.08, S8.09, S8.05, S8.01)
- U8.02 - Working in a team requires effective communication, clear responsibilities, and attention to interpersonal relationships (S8.09, S8.04, S8.05)
- U8.04 - Creative ideas and technical solutions must be communicated in a clear and concise manner (S8.12)
- U8.05 - How people present themselves affects how their work is received (S8.13, S8.12)
- U9.01 - Computer science and information technology careers offer creative job opportunities for individuals with a wide variety of backgrounds and goals (K9.05, K9.04)

Knowledge

Students will:

- K1.01 - Describe the role of creativity in designing an attractive, functional, and accessible graphical user interface (U1.01)
- K2.04 - Identify layers of abstraction used in programming languages and describe advantages and disadvantages inherent in working at a high level of abstraction (U2.02)

- K2.10 - Describe the model-view-controller pattern and relate to a particular software solution (U2.05, U2.06)
- K4.02 - Recognize that a solution to one problem, such as a particular sorting or optimization task, can be used to solve seemingly dissimilar problems (U2.06, U4.01, U4.04)
- K4.05 - Recognize common patterns employing variables, including value accumulation, list aggregation, and iteration across the elements of a collection (U4.01, U5.04)
- K4.06 - Recognize events and event-handlers implied by a user interface (U4.01, U5.02)
- K5.01 - Describe ways to identify the existence and location of errors in software (U5.03)
- K5.02 - Describe the role that functions play in developing software (U4.01, U5.01, U5.04, U5.06)
- K5.03 - Distinguish among a variety of educational and reference resources related to code libraries (U2.05, U5.01, U5.07)
- K7.11 - Describe examples in which computation has or will create new societal phenomena and human capabilities to perceive and act upon our environment (U7.03)
- K8.01 - Describe a version control system
- K9.04 - Describe career-oriented opportunities to use computational skills to positively affect people's lives (U9.01)
- K9.05 - Identify fields of computing careers (U9.01)

Skills

Students will:

- S1.01 - Create a visual artifact of a computational process, such as an image or screenshot of a user interface or web page (U1.01)
- S1.04 - Create a graphical user interface (U1.01, U5.05)
- S1.05 - Analyze a computational artifact for usability with a specific audience in mind (U1.02, U5.05, U7.04)
- S4.01 - Communicate a design for a program using natural language, diagrams, and pseudocode (U5.02)
- S5.01 - Design a program by breaking a large plan into smaller modules (U1.03, U5.02, U5.06)
- S5.03 - Extend, or apply to new purpose, code from other people (U1.03, U2.04, U2.06, U4.04, U5.01)
- S5.04 - Explore and use documentation and public information to extend the student's own knowledge of a programming language or to achieve a computational approach to solve a problem (U2.04, U5.01)
- S5.05 - Use the tools incorporated in an IDE and/or higher-level language to create original programming solutions (U5.01)
- S5.06 - Evaluate programs written by others for readability (U5.08)
- S5.07 - Create a program by incrementally writing and testing modular code (U1.03, U5.03)
- S5.08 - Identify appropriate boundary conditions for testing a program (U4.03, U5.04)
- S5.09 - Improve readability, efficiency, or correctness of code from other people (U4.03, U5.03, U5.08)
- S5.10 - Document a software product using appropriate in-line comments (U5.08)
- S5.11 - Encapsulate a set of related statements in a function or procedure (U2.02, U2.05, U5.02, U5.06)

- S8.01 - Document a software development process (U8.01)
- S8.02 - Document a software product using high-level documentation (U5.07)
- S8.04 - Effectively manage a project, including planning and time management, team norming and load balancing, file/revision management, and documentation (U8.02)
- S8.05 - Collaborate effectively with others when managing a project (U8.01, U8.02)
- S8.07 - Compare to select from among several tools available for collaboration
- S8.08 - Collaborate when programming (U8.01)
- S8.09 - Collaborate when presenting (U8.01, U8.02)
- S8.11 - Distinguish sources of information and comment on the information's reliability and intended audience (U5.01)
- S8.12 - Communicate an idea for a product that solves a problem or expresses creativity (U1.01, U8.04, U8.05)
- S8.13 - Communicate the progress on a project, including accomplishments and next steps (U8.05)

Essential Questions

- ➢ Q1 – How does abstraction make the software development process easier?
- ➢ Q2 – What are the practices that lead to effective collaboration?
- ➢ Q3 – What role does creativity play in algorithmic programming?
- ➢ Q4 – How is computing affecting the way we live our lives?
- ➢ Q5 – How will computing change our world?

Day-by-Day Plans

Activity 1.5.1	Human-Computer Interaction (1 day)
Activity 1.5.2	The API for the Tkinter Canvas (2 days)
Activity 1.5.3	The MVC Pattern with Tkinter (2 days)
Problem 1.5.4	Design a Python GUI (5 days)

Time: 10 days

For time compensation, Activities 1.5.2 through 1.5.4 can be largely omitted. If Activity 1.5.2 through Problem 1.5.4 are omitted, all CS Principles Learning Objectives are still covered elsewhere in this course. However, these three activities cover hexadecimal representation, programming paradigms, scoping, and design patterns. These topics are not repeated elsewhere in this course. The slides corresponding to some or all of these topics can be incorporated by the teacher opting to skip the use of Tkinter.

While still aiming to be accessible to all PLTW students, this lesson aims to open the ceiling for students at the highest level of achievement.

Day 1:

- Present or have students view 1.5.1 HCI.pptx.
- The slide notes are intended as a script for you or your students to use to accompany the slides.
- Guide students through Activity 1.5.1 HCI.
- Assign conclusion questions.

Days 2-3:

- Distribute copies of the 1.5 Reference Card.
- Present the beginning of 1.5.2 Tkinter.pptx. Show the result of 1.5.2 Tkinter_tutorial.py to familiarize students with the widgets.
- Guide students through Activity 1.5.2 Tkinter Canvas.
- Two points to note concerning Tkinter:
    - ➢ If there is an error in a program that creates a Tk() window, that failed window will appear the next time you successfully reach mainloop() with a program. So you'll get a bunch of Tk windows opening, only one of which is the current program.
    - ➢ A bug in Tkinter affects code using create_image(), a method introduced in Step 8d. If the sequence img = PIL.Image.open(filename)
      tkimg = PIL.ImageTk.PhotoImage(img)
      sprite = canvas.create_image(50, 50, image=tkimg)
      fails because the filename doesn't exist or the directory path is incorrect, subsequent execution with a valid filename will raise the error TclError: image "pyimage2" doesn't exist. Commenting out the call to create_image() and executing once will resolve it. Once a program successfully reaches Tkinter.Tk().mainloop(), all the Tk instances from previous failed trials will appear. Then the code above (with a valid filename) will work correctly.

Days 4-5:

- Gather and/or discuss student work from Activity 1.5.2.
- Present the rest of 1.5.2 Tkinter.pptx.
- Present 1.5.3 MVCTkinter.
- Guide students through Activity 1.5.3 Tkinter GUI.
- For each part of this activity, provide the following guidance:
  Starter program demonstration
  Walkthrough presentation
  Solution program demonstration
    - ➢ Do not distribute the solution program (labeled AK for "answer key") to students.
- The support materials for Part I are as follows.
    - ➢ 1.5.3.PY radius_changer.py (starter code)
    - ➢ 1.5.3a Walkthrough1.pptx
    - ➢ 1.5.3.PY.AK position_changer.py (answer key)
- The support materials for Part II are as follows.
    - ➢ 1.5.3.PY color_string_changer.py (starter code)
    - ➢ 1.5.3b Walkthrough2.pptx
    - ➢ 1.5.3.PY.AK color_changer.py (answer key)
- The support materials for Part III are as follows.
    - ➢ 1.5.3.PY canvas_circle_art.py (starter code)
    - ➢ 1.5.3c Walkthrough3.pptx
    - ➢ 1.5.3.PY.AK canvas_rectangle_art.py (answer key, standard)
    - ➢ 1.5.3.PY.AK canvas_shape_art.py (answer key, moderate difficulty)
    - ➢ 1.5.3.PY.AK canvas_recolor_art.py (answer key, higher difficulty)

- The support materials for Part IV are as follows.
    - 1.5.3.PY bouncing_ball.py (starter code)
    - 1.5.3d Walkthrough4.pptx and 1.5.3e Trigonometry.pptx (optional, for differentiation)
    - 1.5.3.PY.AK wrapping_ball.py (answer key)
    - 1.5.3.PY.AK asteroids.py (optional)

Days 6-10:

- Gather and/or discuss student work from Activity 1.5.3
- Introduce Problem 1.5.4 GUI Design
- Support team building and problem solving as students work on Problem 1.5.4.
- Provide students an opportunity to present their work.
- Assist in assembling artifacts into the beginnings of a portfolio. Consider elements of both a showcase portfolio (displaying best work) and a growth portfolio (displaying a progression of skill and sophistication). Written reflection by the student is encouraged to be incorporated into the portfolio.
- Problem 1.5.4 may be used for the College Board's "Create" Performance Assessment Task.

Unit 2.1 The Internet and the Web

In this unit, students will understand the Internet as a set of computers exchanging bits, in the form of packets, and will identify the components of their digital footprint found within the exchange of this information. Students will evaluate a favorite website by comparing the designs, strengths, and weaknesses of the site, in addition to evaluating sites found by their peers. In this context, students will understand and evaluate the content and flow of data when browsing the web, and will compare results from different search engines, simultaneously refining their search techniques to meet a specific objective. Students assess the trustworthiness of web-based media and consider the data flow that permits targeted advertisements, also learning to employ appropriate tools to explore the hierarchical nature of DNS and IP. Students identify ways that a web developer's decisions affect the user and ways that the user's decisions impact society. The tree structure of web documents is introduced alongside HTML and CSS and students will understand how privacy and authentication on the Internet are produced by paired key encryption.

Understandings

- U1.01 - Computing fosters creative expression, sometimes resulting in artifacts. (K1.01) (S1.01)
- U1.02 - Computational artifacts can be evaluated. (S1.05)
- U1.03 - Programming is a creative endeavor.
- U2.01 - Binary sequences represent digital data. (K2.03, K2.01) (S2.05)
- U2.02 - Computing relies on layers of abstraction in software. (K2.04)
- U2.03 - Computing relies on abstractions of hardware represented with software. (K2.07)
- U3.08 - Ethical and societal issues are raised by the impact of Big Data and require attention from many stakeholders.
- U4.01 - Programs implement algorithms to solve problems. (S6.08)

- U5.03 - Creating solutions with computation requires a persistent, iterative problem-solving approach. (K5.01)
- U5.05 - The user interface of a piece of software can greatly affect how it is used. (K7.06) (S1.05)
- U6.01 - Networked and mobile computing rely on various protocols to provide services. (K6.03, K6.06, K6.07, K6.04, K6.05, K6.16) (S6.12, S6.03, S6.06, S6.05)
- U6.02 - A variety of languages are used for web programming, with both overlapping and complementary purposes. (K6.07, K6.01) (S1.02, S6.02, S6.01)
- U6.03 - The Internet facilitates collaboration. (K7.07, K7.11)
- U6.04 - Device-to-device communication through the Internet passes through a non-unique route. (K6.08, K6.09) (S6.12)
- U6.06 - Maintaining a safe presence on the Internet requires attention and knowledge.
- U6.07 - Cybersecurity depends on hardware and software components, including cryptography. (K6.16, K6.13)
- U7.01 - Mobile and networked computing have transformed commerce, social interactions, news sourcing and dissemination, and culture. (K7.02, K7.07, K7.15, K7.11)
- U7.04 - Making information accessible to all people requires attention from a variety of stakeholders. (K7.07, K7.14, K7.06) (S1.05)
- U7.05 - New opportunities for human creativity and innovation exist because of networked, mobile, and embedded computing. (K1.01)
- U7.07 - Networked infrastructure affects and is affected by commercial and governmental structures and policies.
- U7.08 - Scalability is an important consideration for distributed solutions.
- U8.01 - Computing artifacts and programs can be higher quality as a result of collaboration. (S8.08, S8.09)
- U8.02 - Working in a team requires effective communication, clear responsibilities, and attention to interpersonal relationships. (S8.09)
- U9.02 - Parallel computing is a quickly evolving field relevant to hardware, software, and users.

Knowledge

Students will . . .

- K1.01 - Describe the role of creativity in designing an attractive, functional, and accessible graphical user interface. (U1.01, U7.05)
- K2.03 - Describe layers of abstraction that help people represent and use data. (U2.01)
- K2.04 - Identify layers of abstraction used in programming languages and describe advantages and disadvantages inherent in working at a high level of abstraction. (U2.02)
- K2.07 - Describe an abstraction of hardware. (U2.03)
- K2.12 - Identify the relationship among nodes in a tree, as applied to the DOM in a web page.
- K5.01 - Describe ways to identify the existence and location of errors in software. (U5.03)
- K5.03 - Distinguish among a variety of educational and reference resources related to code libraries.
- K6.01 - Identify syntactic elements of HTML and CSS. (U6.02)

- K6.03 - Describe the hierarchical nature of the domain name system and IPv4 and IPv6 numbering systems. (U6.01)
- K6.04 - Describe the dependence of DNS on certificate authorities. (U6.01)
- K6.05 - Describe abstractions that enable the Internet to function, including IP addressing and domain name service. (U6.01)
- K6.06 - Describe the purpose and general nature of protocols that enable the Internet to function, including DNS, TCP/IP, SMTP, and HTTP protocols. (U6.01)
- K6.07 - Describe the governance of the Internet and the organizations that develop and maintain relevant standards. (U6.01, U6.02)
- K6.08 - Identify examples of redundancy and autonomy in the physical and software systems of the Internet. (U6.04)
- K6.09 - Explain how redundancy and autonomy make the systems of the Internet scalable. (U6.04)
- K6.13 - Explain an encryption system. (U6.07)
- K6.16 - Describe the combination of encryption protocols with other protocols to provide secure transfer of information. (U6.01, U6.07)
- K7.02 - Identify the decade in which milestones occurred in the development of computing and the Internet. (U7.01)
- K7.05 - Describe what metadata contain and how they can be used.
- K7.06 - Distinguish content from style and explain how accessibility requires a separation of these concerns. (U5.05, U7.04)
- K7.07 - Analyze access to the Internet and to data among people in different countries, and connect to measures of health and wealth. (U6.03, U7.01, U7.04)
- K7.11 - Describe examples in which computation has or will create new societal phenomena and human capabilities to perceive and act upon our environment. (U6.03, U7.01)
- K7.14 - Contrast the patterns of inequity characterizing the personal computing revolution and the current mobile computing revolution. (U7.04)
- K7.15 - Characterize the size, cost, and speed of computational processing as changing exponentially. (U7.01)
- K9.06 - Describe how computing is connected to innovations in other fields.
- K9.07 - Describe how specific career fields, e.g., biology and marketing, have changed profoundly as a result of automated collection and processing of data.

Skills

Students will . . .

- S1.01 - Create a visual artifact of a computational process, such as an image or screenshot of a user interface or web page. (U1.01)
- S1.02 -Design, create, and publish a basic web page referencing a style sheet. (U6.02)
- S1.05 - Analyze a computational artifact for usability with a specific audience in mind. (U1.02, U5.05, U7.04)
- S2.05 - Calculate time, file size, and bandwidth given two of the three. (U2.01)
- S6.01 - Connect the rendering of a web page with corresponding elements of HTML and CSS source code and with client browser settings. (U6.02)
- S6.02 - Identify HTML, CSS, JavaScript, PHP, or SQL as an appropriate language for a particular task. (U6.02)

- S6.03 - Use appropriate tools to manage files on a server permitting FTP. (U6.01)
- S6.05 - Use appropriate tools to observe bandwidth and latency. (U6.01)
- S6.06 - Analyze a URL to identify protocol, the host and domain names, the directory path, the filename, and the query string content, and describe the information implied about the nature of the requested content and its publisher. (U6.01)
- S6.08 - Contrast search results provided by competing search algorithms to identical queries. (U4.01)
- S6.12 - Use appropriate tools to observe IP addressing, IP routing, and DNS resolution. (U6.01, U6.04)
- S8.08 - Collaborate when programming. (U8.01)
- S8.09 - Collaborate when presenting. (U8.01, U8.02)
- S8.10 - Present original research and research from literature.
- S8.11 - Distinguish sources of information and comment on the information's reliability and intended audience.

Essential Questions

- Q1 - How does the Internet work?
- Q2 - How can we protect ourselves, our privacy, and our assets when working on the Internet?
- Q3 - How has the Internet affected society?

Day-by-Day Plans

Activity 2.1.1	The Rise of the Internet (2 days)
Activity 2.1.2	Your Favorite Web Page (1 day)
Activity 2.1.3	Protocols and Bandwidth (2 days)
Project 2.1.4	HTML and CSS (3 days)
Activity 2.1.5	Secure Protocols (1 day)

Time: 9 days

Days 1–2

- Guide students through Activity 2.1.1 steps 1–5.
- Distribute Packet handouts.
- Guide students through the remainder of Activity 2.1.1.
- Facilitate student research and presentation of research.
- Assign Conclusion questions.
- Present *2.1.1 Cyber Ethics Cyber Law* and have students turn in *2.1.1 The Pledge*. Keep a copy of their signed pledge to demonstrate that you informed students of the importance of ethical behavior on the Internet. Because students are being given access to a server outside the school filter, it is important to ensure that they are aware that unethical actions can have severe consequences.
- The *Lesson 2.1 Supplement: Creating a Cloud9 Workspace* can be used at any point prior to Activity 2.1.3. You may wish to have students identify an available Cloud9 username for your approval prior to the activity.

Day 3

- Review conclusion questions from 2.1.1.

- Model usage of Firebug for gathering information. Focus on use of the cookies tab, and use the HTML and CSS tags to explore what code connects to which visual features of a web page.
- Guide students through Activity 2.1.2.
- At step 30, you may wish to demonstrate use of Google Trends to the class, as opposed to having them do it on their own, to avoid reaching the limit that Google places on queries by IP address.
- Assign Conclusion questions.

Days 4–6

- Have students complete the *Lesson 2.1 Supplement: Creating Cloud9 Workspace*. This supplement is provided in the Activity 2.1.3 resources.
- Guide students through *Activity 2.1.3 Protocols and Bandwidth*. In step 19, you might want to point your students to a specific article or choice among articles from the ACM TechNews archive. An example is: http://www.usatoday.com/story/tech/2013/12/01/tech-firms-counter-nsa-data/3495995/

Days 7–8

- Model/review how to enter a Cloud9 workspace and interact with the Bash shell.
- Guide students through Activity 2.1.4.
- Assign conclusion questions.

Day 9

- Prepare a Google document or spreadsheet with a row for each student to publish the public key and an encrypted message. Share the document with all students. If your students are new to collaborating through a Google Doc or Google Sheet, provide a quick tour of the interface.
- Guide students through *Activity 2.1.5 Secure Protocols*.

Unit 2.2 Shopping and Social on the Web

The objective of this unit is for students to understand the role of client-side code, server-side code, and their interaction with databases in delivering interactive web-based content. Ideally, CS students collaborate with art students and with clubs and organizations on campus to publish content on the Web. Students are provided with JavaScript and PHP code and can access an SQL database from a secure shell command line as well as through PHP. Students compare languages they have encountered so far to generalize the concepts of sequencing instructions, selection of instructions by conditionals, iteration, and the common roles of variables. Students explore and compare career paths within computing.

Understandings

- U1.01 - Computing fosters creative expression, sometimes resulting in artifacts. (K1.01) (S1.01, S8.12)
- U1.02 - Computational artifacts can be evaluated. (S1.05)
- U1.03 - Programming is a creative endeavor. (S1.03, S5.03)

- U2.06 - The solution to one problem can be applied to another seemingly unrelated problem by identifying and reusing a pattern. (K2.10, K4.02) (S5.03)
- U3.01 - Data can be structured to facilitate use. (K3.02, K2.12, K7.05)
- U3.06 - The size of a data set affects how the data can be used. (K3.05)
- U3.07 - Collecting and managing data raises technical issues regarding storage, access, durability, privacy, and security. (K3.05, K7.05)
- U4.01 - Programs implement algorithms to solve problems. (K5.02, K4.05, K4.06, K4.02)
- U4.04 - A given algorithmic problem with standard solutions can be applied in diverse contexts. (K4.02) (S5.03)
- U5.02 - Solutions in a programming language are created by breaking a problem apart into component problems. (K4.06) (S4.01)
- U5.03 - Creating solutions with computation requires a persistent, iterative problem-solving approach. (K5.01) (S5.09)
- U5.04 - Programming requires an understanding of mathematical operations and data abstractions. (K5.02, K3.03, K4.05)
- U5.05 - The user interface of a piece of software can greatly affect how it is used. (K7.06) (S1.05)
- U5.06 - Functions with arguments make code modular and reusable. (K5.02)
- U5.07 - Programmers create high-level documentation to communicate the purpose and function of their code. (K5.03)
- U5.08 - Programmers must prioritize making their code well-documented and readable for it to be maintained. (S5.09)
- U6.01 - Networked and mobile computing rely on various protocols to provide services. (S6.03)
- U6.02 - A variety of languages are used for web programming with both overlapping and complementary purposes. (K6.01, K6.02) (S1.03, S6.10, S6.02)
- U6.03 - The Internet facilitates collaboration.
- U6.05 - The information and processing power on any networked device can be accessed by potentially hostile parties.
- U6.06 - Maintaining a safe presence on the Internet requires attention and knowledge. (S6.09, S6.07)
- U7.04 - Making information accessible to all people requires attention from a variety of stakeholders. (K7.06) (S1.05)
- U7.05 - New opportunities for human creativity and innovation exist because of networked, mobile, and embedded computing. (K1.01)
- U7.08 - Scalability is an important consideration for distributed solutions. (K3.05)
- U8.01 - Computing artifacts and programs can be higher quality as a result of collaboration. (S8.08, S8.09, S8.05)
- U8.02 - Working in a team requires effective communication, clear responsibilities, and attention to interpersonal relationships. (S8.09, S8.04, S8.05)
- U8.03 - Collaboration allows communities to create software that can impact people’s lives. (K8.01) (S8.08, S8.03)
- U8.04 - Creative ideas and technical solutions must be communicated in a clear and concise manner. (S8.12)

- U8.05 - How people present themselves affects how their work is received. (S8.10, S8.13, S8.12)
- U9.01 - Computer science and information technology careers offer creative job opportunities for individuals with a wide variety of backgrounds and goals. (K9.05, K9.04)
- U9.03 - Computational thinking boosts most career paths. (K9.07, K9.04)

Knowledge

Students will...

- K1.01 - Describe the role of creativity in designing an attractive, functional, and accessible graphical user interface. (U1.01, U7.05)
- K2.03 - Describe layers of abstraction that help people represent and use data.
- K2.04 - Identify layers of abstraction used in programming languages and describe advantages and disadvantages inherent in working at a high level of abstraction.
- K2.10 - Describe the model-view-controller pattern and relate to a particular software solution. (U2.06)
- K2.12 - Identify the relationship among nodes in a tree, as applied to the DOM in a Web page. (U3.01)
- K3.02 - Describe the structure of a relational database. (U3.01)
- K3.03 - Describe how mathematical functions and algorithms can be applied to a data set to construct a derived data set. (U5.04)
- K3.05 - Describe the reasons for the rise in non-relational databases. (U3.06, U3.07, U7.08)
- K4.02 - Recognize that a solution to one problem, such as a particular sorting or optimization task, can be used to solve seemingly dissimilar problems. (U2.06, U4.01, U4.04)
- K4.05 - Recognize common patterns employing variables, including value accumulation, list aggregation, and iteration across the elements of a collection. (U4.01, U5.04)
- K4.06 - Recognize events and event-handlers implied by a user interface. (U4.01, U5.02)
- K5.01 - Describe ways to identify the existence and location of errors in software. (U5.03)
- K5.02 - Describe the role that functions play in developing software. (U4.01, U5.04, U5.06)
- K5.03 - Distinguish among a variety of educational and reference resources related to code libraries. (U5.07)
- K6.01 - Identify syntactic elements of HTML and CSS. (U6.02)
- K6.02 - Identify syntactic elements of JavaScript, PHP, and SQL. (U6.02)
- K7.05 - Describe what metadata contain and how they can be used. (U3.01, U3.07)
- K7.06 - Distinguish content from style and explain how accessibility requires a separation of these concerns. (U5.05, U7.04)
- K8.01 - Describe a version control system. (U8.03)
- K9.04 - Describe career-oriented opportunities to use computational skills to positively affect people's lives. (U9.01, U9.03)
- K9.05 - Identify fields of computing careers. (U9.01)
- K9.07 - Describe how specific career fields, e.g., biology and marketing, have changed profoundly as a result of automated collection and processing of data. (U9.03)

Skills

Students will...

- S1.01 - Create a visual artifact of a computational process, such as an image or screenshot of a user interface or web page. (U1.01)
- S1.03 - Create and publish a basic web page containing JavaScript. (U1.03, U6.02)
- S1.05 - Analyze a computational artifact for usability with a specific audience in mind. (U1.02, U5.05, U7.04)
- S4.01 - Communicate a design for a program using natural language, diagrams, and pseudocode. (U5.02)
- S5.02 - Analyze and test code from other people. (U4.03)
- S5.03 - Extend, or apply to new purpose, code from other people. (U1.03, U2.06, U4.04)
- S5.09 - Improve readability, efficiency, or correctness of code from other people. (U4.03, U5.03, U5.08)
- S6.02 - Identify HTML, CSS, JavaScript, PHP, or SQL as an appropriate language for a particular task. (U6.02)
- S6.03 - Use appropriate tools to manage files on a server permitting FTP. (U6.01)
- S6.07 - Describe the digital footprint left behind and analyze the implications for privacy that result from various actions conducted in a web browser, including DNS requests, content requests, third-party content requests, cookies, and cached content. (U6.06)
- S6.09 - Identify vulnerabilities to social engineering, including phishing and the delivery of viruses through various file types, and respond appropriately. (U6.06)
- S6.10 - Identify an appropriate language for serving a particular purpose in web serving, including tasks appropriate to client-side and server-side scripting. (U6.02)
- S8.03 - Use a version control system. (U8.03)
- S8.04 - Effectively manage a project, including planning and time management, team norming and load balancing, file/revision management, and documentation. (U8.02)
- S8.05 - Collaborate effectively with others when managing a project. (U8.01, U8.02)
- S8.08 - Collaborate when programming. (U8.01, U8.03)
- S8.09 - Collaborate when presenting. (U8.01, U8.02)
- S8.10 - Present original research and research from literature. (U8.05)
- S8.12 - Communicate an idea for a product that solves a problem or expresses creativity. (U1.01, U8.04, U8.05)
- S8.13 - Communicate the progress on a project, including accomplishments and next steps. (U8.05)

Essential Questions

- ➢ Q1 – How does the Internet work?
- ➢ Q2 – How can we protect ourselves, our privacy, and our assets when working on the Internet?
- ➢ Q3 – How has the Internet affected society?

Day-by-Day Plans

Activity 2.2.1 HTML5 and JavaScript (3 days)

	(U1.01, U1.03, U4.01, U5.04, U5.06, U6.01, U6.02) (K1.01, K5.03, K5.02, K6.01, K2.04, K9.05, K6.02, K7.06, K2.12, K4.06) (S1.03, S1.01, S1.05, S8.08, S6.10, S6.02, S6.03, S5.03)
Activity 2.2.2	Introducing PHP (3 days) (U4.03, U4.01, U5.04, U5.07, U5.08, U6.01, U6.02) (K5.02, K5.01, K6.02, K4.05, K4.06) (S1.01, S6.09, S8.08, S6.10, S6.02, S6.03, S5.03, S5.02, S6.07)
Activity 2.2.3	Databases and SQL (2 days) (U7.08, U4.01, U4.04, U3.07, U3.06, U3.01, U5.04, U2.06, U6.02) (K2.03, K2.04, K9.05, K6.02, K3.05, K3.03, K3.02, K2.10, K7.05, K4.02) (S8.08, S6.02, S6.03, S5.03)
Project 2.2.4	Dynamic Data-Driven Design (5 days) (U1.01, U1.03, U1.02, U9.01, U4.01, U7.05, U8.04, U8.05, U8.03, U8.01, U5.05, U5.04, U5.03, U5.02, U8.02, U6.01) (K8.01, K1.01, K9.04) (S1.01, S1.05, S8.08, S8.13, S8.12, S8.09, S6.10, S6.02, S6.03, S5.09, S5.03, S8.04, S8.05, S8.03, S4.01)
Activity 2.2.5	Career Fields of CS and IT (1 day) (U7.04, U9.01, U8.03, U9.03, U6.05, U6.03, U6.06) (K1.01, K9.07, K9.05, K9.04) (S8.10)

Time: 14 days

Days 1–3

- Present 2.2.1 HTML5 and JS to familiarize your students with W3C and some JavaScript syntax.
- Guide students through steps 1–3 of the activity.
- Instruct from the front of the classroom on JavaScript as necessary.
- Guide students through steps 4–13 of the activity.
- Distribute Scrum poker cards.
- Guide students through the remainder of the activity.
- Assign conclusion questions.

Days 4–6

- Review conclusion questions from Activity 2.2.1.
- Review for your students the uploading of necessary files to the server according to step 5 in Activity 2.2.2.
- Review accessing an individual student’s site according to step 6.
- Circulate and guide students through the activity.
- Assign conclusion questions.

Days 7–8

- Review conclusion questions from Activity 2.2.2.

- Demonstrate a few simple queries on the shoe database from the front of the class.
- Circulate and guide students through Steps 1–19 of Activity 2.2.3.
- In front of the class, highlight key parts of 2.2.3index.php.
- Circulate and guide students through Steps 20–29.
- Provide direct instruction on Normalization.
- Circulate and guide students through the rest of Activity 2.2.3.
- Assign conclusion questions.

Days 9–13

- Review conclusion questions from Activity 2.2.3.
- Review Problem 2.2.4 rubric and CSP presentation rubrics with students in front of the class.
- Explain the two options for the problem as in Step 3.
- Demonstrate Scrum Poker.
- Circulate and guide students through the rest of the Problem.
- Assign conclusion questions as homework. Note that questions with word counts are meant to reinforce protocol for the AP CSP portfolio assessment.

Day 14

- Present 2.2.5 CS Careers.
- Circulate and guide students through Activity 2.2.5.
- Facilitate presentations.
- Assign conclusion questions.

Unit 2.3 Cybersecurity and Cryptography

The goal of this unit is for students to personally invest in maintaining online security and to improve their personal cybersecurity hygiene. Students focus on cybersecurity from the perspectives of the user, the software developer, businesses, the nation, and the citizen. In the team competition at the end of the lesson, students explore parallel strands in encryption and security. Encryption is used as a route to explore the efficiency of algorithms and how the time required for an algorithm to execute can depend on its input.

Understandings

- U3.07 - Collecting and managing data raises technical issues regarding storage, access, durability, privacy, and security
- U3.08 - Ethical and societal issues are raised by the impact of Big Data and require attention from many stakeholders
- U4.01 - Programs implement algorithms to solve problems (S6.08)
- U4.02 - Algorithms can be analyzed for efficiency, and appropriate algorithms can be selected based upon efficiency (K4.04)
- U4.04 - A given algorithmic problem with standard solutions can be applied in diverse contexts
- U5.01 - Creating solutions with computation requires exploring the tools available, selecting an appropriate tool, and gaining expertise with the tool

- U5.03 - Creating solutions with computation requires a persistent, iterative problem-solving approach
- U5.04 -Programming requires an understanding of mathematical operations and data abstractions (K3.03)
- U6.03 - The Internet facilitates collaboration
- U6.04 - Device-to-device communication through the Internet passes through a non-unique route
- U6.05 - The information and processing power on any networked device can be accessed by potentially hostile parties (K6.14, K6.11)
- U6.06 - Maintaining a safe presence on the Internet requires attention and knowledge (K6.14, K6.11, K6.12) (S6.09, S6.11, S6.07, S6.04)
- U6.07 - Cybersecurity depends on hardware and software components, including cryptography (K6.16, K6.11, K6.10, K6.13, K6.12) (S6.11)
- U7.01 - Mobile and networked computing have transformed commerce, social interactions, news sourcing and dissemination, and culture (S7.02)
- U7.02 - Computing is having profound impacts on individual privacy (S7.02)
- U7.04 - Making information accessible to all people requires attention from a variety of stakeholders
- U7.07 - Networked infrastructure affects and is affected by commercial and governmental structures and policies (S7.02)
- U8.02 - Working in a team requires effective communication, clear responsibilities, and attention to interpersonal relationships
- U9.03 - Computational thinking boosts most career paths (K9.04, K9.06, K9.02, K3.04)

Knowledge

Students will:

- K3.03 - Describe how mathematical functions and algorithms can be applied to a data set to construct a derived data set (U5.04)
- K3.04 - Identify the relationship between parallel computation and computationally intensive tasks like simulation, modeling, and analysis of large data sets (U9.03)
- K4.04 - Estimate the time for an algorithm to operate on a data set, given the algorithm's running time on another data set and the algorithm's complexity in big-O notation (U4.02)
- K6.07 - Describe the governance of the Internet and the organizations that develop and maintain relevant standards
- K6.10 - Describe the mechanism of any particular cyber attack (U6.05, U6.06, U6.07)
- K6.12 - Identify measures that contribute to an effective defense from any particular cyber attack (U6.06, U6.07)
- K6.13 - Explain an encryption system (U6.07)
- K6.14 - Identify the roles of software developers, government, industry, employees, standards bodies, consumers, and citizens in securing information and processing power (U6.05, U6.06)
- K6.16 - Describe the combination of encryption protocols with other protocols to provide secure transfer of information (U6.07)
- K9.01 - Describe codes of ethics and professional conduct for cybersecurity professionals

- K9.02 - Describe the impact that computing has had in the social sciences, geography, and civics (U9.03)
- K9.04 - Describe career-oriented opportunities to use computational skills to positively affect people's lives (U9.03)
- K9.05 -Identify fields of computing careers
- K9.06 - Describe how computing is connected to innovations in other fields (U9.03)

Skills

Students will:

- S2.06 - Consider implications of converting data from one representation to another, for example noise or compression
- S6.04 - Analyze the security of situations in which a user is prompted to follow a link, download content, or provide information in email and Web forms (U6.06)
- S6.06 - Analyze a URL to identify protocol, the host and domain names, the directory path, the filename, and the query string content, and describe the information implied about the nature of the requested content and its publisher
- S6.07 - Describe the digital footprint left behind and analyze the implications for privacy that result from various actions conducted in a Web browser, including DNS requests, content requests, third-party content requests, cookies, and cached content (U6.06)
- S6.08 - Contrast search results provided by competing search algorithms to identical queries (U4.01)
- S6.09 - Identify vulnerabilities to social engineering, including phishing and the delivery of viruses through various file types, and respond appropriately (U6.06)
- S6.11 - Use appropriate tools and techniques to implement defensive cyber measures (U6.06, U6.07)
- S7.02 - Work with a group to agree on a policy or protocol (U7.01, U7.02, U7.07)

Essential Questions

- ➢ Q1 - What is the nature of attack and defense in cybersecurity?
- ➢ Q2 - How can we protect ourselves, our privacy, and our assets when working on the Internet?
- ➢ Q3 - How has the Internet affected society?

Day-by-Day Plans

Activity 2.3.1 - The Vulnerable User (2days)
(U7.02, U7.07, U9.03, U6.04, U6.05, U6.06, U6.07) (K6.14, K6.11, K6.12) (S6.09, S6.11, S2.06, S6.06, S6.07, S6.04)

Activity 2.3.2 - Security by Encryption (1 day)
(U4.02, U4.01, U4.04, U5.04, U6.07) (K3.04, K3.03, K6.16, K6.10, K6.13, K4.04) (S2.06)

Activity 2.3.3 – Security and Liberty (2 days)
(U7.07, U7.04, U7.01, U3.08, U3.07, U6.03, U6.06) (K6.07, K9.04, K9.06, K9.02) (S7.02, S6.08, S6.07)

Project 2.3.4 - The Heist (3 days)
(U4.01, U5.04, U5.01, U5.03, U8.02, U6.07) (K9.01, K9.05, K6.14, K6.11, K6.13, K6.12)

Time: 8 days

Days 1-2:

- Introduce the Lesson with Lesson Preface and the Essential Questions.
- Present or have students view 2.3.1 Cyberhygiene.pptx.
- Guide students through Activity 2.3.1 The Vulnerable User.docx, during which they will work with 2.3.1 TypesOfMalware.pptx, 2.3.1 SocialEngineering.pptx, 2.3.1 ProtectingIdentity.pptx, and 2.3.1 UseSafeSoftware.pptx.
- Assign conclusion questions as homework.
- Assign *Blown to Bits* reading pages 48-72 and 187-193 for Steps 9-10 of Activity 2.3.3.

Day 3:

- Present 2.3.2 Complexity.pptx. An optional presentation, 2.3.2 RSA.pptx is provided for students who are interested.
- Circulate as students work on Activity 2.3.2 SecurityByEncryption.docx. Decide how you will handle the optional Step 9 and the checkpoint in Step 10.
- Steps 20-25 and conclusion questions 1-3 can be completed as homework if desired. Steps 21, 23, and 25 can be omitted to lower the rigor and reading level.

Day 4:

- Review and discuss student work from Activity 2.3.2. Focus on student thinking in Step 12 in which they compared the algorithms. Steps 22 and 23 assess how well students understand the relationship between time efficiency, cybersecurity, and complexity.
- Introduce Activity 2.3.3 Security and Liberty. In step 2b, you will lead a class discussion to collect all groups' contributions in a single list. To that list, add the items shown in bold on the answer key if they were missing from the class's list. Place a star next to those items for all groups to consider in Step 3.
- For prompts which require individual writing, like Step 3b or Step 7, you might have students do prewriting in class in the sequence of the activity but ask them to finish the writing outside of class.

Day 5:

- Continue guiding discussion in Activity 2.3.3.
- Step 13 of Activity 2.3.3 offers two options. The second option is a Research Performance Task and is much more time consuming than the first option. A similar Research Performance Task is offered as an option in each Unit, listed below. Students should complete the longer option as a Research Performance Task in at least one unit in the course. This corresponds to the AP Computer Science *Investigate* Performance Task. Students submitting performance tasks for AP Computer Science assessment should complete two such Research Performance Tasks so that they can get teacher guidance on the first one before completing the second one for external evaluation.
- Unit 1: Human-Computer Interaction

- Unit 2: Law Enforcement, Privacy, Democracy
- Unit 3: Impact of Big Data
- Unit 4: Impact of Robotics, Artificial Intelligence, and Simulation
- Establish a timeline for completing Step 13. The deadline for the Research Performance Task is likely to be beyond the end of Unit 2.
- Provide opportunity for students to review each other's work for Conclusion Question 2.

Days 6-8:

- Present 2.3.4 PenetrationTesting.pptx. Introduce Project 2.3.4. Depending on student interest, class time used for the cybersecurity competition could be decreased and/or an extracurricular opportunity could be provided for students who want to dive deeper.

Unit 3.1 Visualizing Data

The goal of this unit is for students to be able to create visualizations to analyze sets of large data and to meaningfully interpret the patterns they uncover. They draw conclusions about themselves from relevant data, including community Census reports, local weather, the economics of their community, and naming trends. At the beginning of the lesson, students weigh societal concerns around the collection and persistence of Big Data. The students learn how to use *Python*® programming language to make useful graphic representations of data, developing from familiar visualizations to more modern visual analyses like scaled-dot or colorized scatter plots of multidimensional data sets. Students are introduced to basic Excel® spreadsheet programming and cell manipulation. A Monte Carlo simulation is used to help students appreciate the meaning of evidence for association between two variables.

Understandings

- U1.02 - Computational artifacts can be evaluated. (S1.06)
- U2.01 - Binary sequences represent digital data. (K2.02, K2.03, K7.08, K2.16)
- U3.01 - Data can be structured to facilitate use. (K3.01) (S3.07)
- U3.03 - Analysis of data can be automated. (K3.03) (S3.05, S3.07, S3.03, S3.08, S5.12)
- U3.04 - Data visualizations are important tools for discovering and communicating knowledge. (S8.06, S3.05, S3.04, S3.03, S3.08)
- U3.05 - The human brain and today's computers have complementary strengths for analyzing data. (S3.04, S3.07, S3.08)
- U3.06 - The size of a data set affects how the data can be used. (K4.04)
- U3.07 - Collecting and managing data raises technical issues regarding storage, access, durability, privacy, and security. (K7.08)
- U3.08 - Ethical and societal issues are raised by the impact of Big Data and require attention from many stakeholders. (K7.09)
- U4.01 - Programs implement algorithms to solve problems. (K2.08, K4.05) (S5.12)
- U4.02 - Algorithms can be analyzed for efficiency, and appropriate algorithms can be selected based upon efficiency. (K4.04) (S4.02)
- U4.04 - A given algorithmic problem with standard solutions can be applied in diverse contexts. (K2.08)
- U5.02 - Solutions in a programming language are created by breaking a problem apart into component problems. (S5.11)

- U5.03 - Creating solutions with computation requires a persistent, iterative problem-solving approach. (S5.09)
- U5.04 - Programming requires an understanding of mathematical operations and data abstractions. (K3.03, K4.05)
- U5.07 - Programmers create high-level documentation to communicate the purpose and function of their code.
- U5.08 - Programmers must prioritize making their code well-documented and readable for it to be maintained. (S5.09, S5.10)
- U6.03 - The Internet facilitates collaboration. (K7.07)
- U6.05 - The information and processing power on any networked device can be accessed by potentially hostile parties.
- U7.01 - Mobile and networked computing have transformed commerce, social interactions, news sourcing and dissemination, and culture. (K7.07) (S8.07)
- U7.02 - Computing is having profound impacts on individual privacy. (K7.09)
- U7.04 - Making information accessible to all people requires attention from a variety of stakeholders. (K7.07, K7.14)
- U7.08 - Scalability is an important consideration for distributed solutions. (K4.04)
- U8.01 - Computing artifacts and programs can be higher quality as a result of collaboration. (S8.08, S8.06)
- U8.02 - Working in a team requires effective communication, clear responsibilities, and attention to interpersonal relationships. (S8.06)
- U9.01 - Computer science and information technology careers offer creative job opportunities for individuals with a wide variety of backgrounds and goals. (K9.05, K9.04)

Knowledge

Students will:

- K1.02 - Describe the role of creativity in producing data visualizations.
- K2.02 - Distinguish continuous and discrete phenomena and identify digital and analog data. (U2.01)
- K2.03 - Describe layers of abstraction that help people represent and use data. (U2.01)
- K2.04 - Identify layers of abstraction used in programming languages and describe advantages and disadvantages inherent in working at a high level of abstraction.
- K2.08 - Distinguish deterministic and randomized models and describe the role of Monte Carlo techniques. (U4.01, U4.04)
- K2.16 - Describe the implications of the limited precision of digital information in applications. (U2.01)
- K3.01 - Relate the categorical or quantitative nature of data to the operations and visualizations that are appropriate. (U3.01)
- K3.03 - Describe how mathematical functions and algorithms can be applied to a data set to construct a derived data set. (U3.03, U5.04)
- K3.04 - Identify the relationship between parallel computation and computationally intensive tasks like simulation, modeling, and analysis of large data sets.
- K3.06 - Identify data visualizations as an important tool for communication.
- K4.05 - Recognize common patterns employing variables, including value accumulation, list aggregation, and iteration across the elements of a collection. (U4.01, U5.04)

- K7.07 - Analyze access to the Internet and to data among people in different countries and connect this data to measures of health and wealth. (U6.03, U7.01, U7.04)
- K7.08 - Discuss societal implications of the persistence and the ease of copying digital information. (U2.01, U3.07)
- K7.09 - Articulate a range of positions on questions related to privacy with respect to Big Data. (U3.08, U7.02)
- K7.14 - Contrast the patterns of inequity characterizing the personal computing revolution and the current mobile computing revolution. (U7.04)
- K9.04 - Describe career-oriented opportunities to use computational skills to positively affect people's lives. (U9.01)
- K9.05 - Identify fields of computing careers. (U9.01)
- K9.06 - Describe how computing is connected to innovations in other fields.
- K9.07 - Describe how specific career fields, e.g., biology and marketing, have changed profoundly as a result of automated collection and processing of data.

Skills

Students will:

- S1.01 - Create a visual artifact of a computational process, such as an image or screenshot of a user interface or web page.
- S1.06 - Analyze a computational artifact for correctness. (U1.02)
- S3.03 - Represent and understand phenomena by identifying relationships among data. (U3.03, U3.04)
- S3.04 - Identify the type of basic visualization appropriate to a particular data set. (U3.04, U3.05)
- S3.05 - Use appropriate tools to construct a scatter plot, histogram, pie chart, or compound bar graph. (U3.03, U3.04)
- S3.07 - Use appropriate tools to create new data sets derived from other data sets. (U3.01, U3.03, U3.05)
- S5.09 - Improve readability, efficiency, or correctness of code from other people. (U4.03, U5.03, U5.08)
- S5.10 - Document a software product using appropriate in-line comments. (U5.08)
- S5.11 - Encapsulate a set of related statements in a function or procedure. (U5.02)
- S5.12 - Create a function to perform a particular calculation from the function's arguments. (U3.03, U4.01)
- S5.13 - Deduce the value of a variable resulting from execution of particular code. (U4.03)
- S8.06 - Collaborate when collecting and analyzing data to answer a question. (U3.04, U8.01, 8.02)
- S8.07 - Compare to select from among several tools available for collaboration. (U7.01)
- S8.08 - Collaborate when programming. (U8.01)
- S8.11 - Distinguish sources of information and comment on the information's reliability and intended audience.

Essential Questions

- ➢ Q1 - How will computation impact fields other than computing itself?
- ➢ Q2 - How will computation impact society?

- Q3 - How can patterns be discovered in data?

Day-by-Day Plans

A3.1.1 Time Series and Trends (3 days)
(U1.02, U9.01, U4.03, U4.01, U3.04, U3.03, U8.01, U5.04, U5.03, U5.08, U8.02) (K2.03, K3.06, K3.01, K3.03) (S1.01, S1.06, S8.08, S5.09, S8.06, S3.04, S3.03, S5.10, S5.11, S5.12, S5.13)

A3.1.2 Issues with Data (2 days)
(U7.02, U7.01, U9.01, U3.08, U3.07, U3.06, U3.05, U6.05) (K9.04, K7.09, K7.08)

A3.1.3 Big Data and Parallel Processing (2 days)
(U7.08, U4.02, U4.01, U4.04, U3.06, U3.04, U3.03, U2.01) (K2.03, K2.04, K3.04, K3.06, K3.01, K3.03, K4.04) (S1.01, S1.06, S8.06, S3.05, S3.04, S3.03, S3.08, S4.02)

A3.1.4 Pie Charts and Bar Graphs (3 days)
(U1.02, U7.04, U9.01, U3.05, U3.04, U6.03) (K1.02, K2.02, K7.07, K9.07, K7.14, K9.05, K9.04, K9.06, K3.06, K3.01) (S1.01, S1.06, S8.07, S8.06, S3.05, S3.04, S8.11)

A3.1.5 Histograms and Distributions (4 days)
(U4.01, U3.05, U3.03, U3.01) (K2.03, K3.01, K3.03, K2.16, K4.05) (S8.06, S3.07, S3.03, S5.12)

Time: 14 days

Day 1:

- Frame the unit as building tools to learn to explore Big Data sets. In Problem 3.2.7, students will explore and analyze a data set to answer questions of their choosing. The artifacts from that problem can be submitted for the CSP portfolio.
- Direct student attention to the Essential Questions for the lesson and the objectives for student understanding with specific skills and knowledge. Show the key terms to remind students of this resource.
- Introduce Activity 3.1.1 Time Series and Trends. Circulate as students work.

Days 2-3:

- Discuss steps and conclusion questions from Activity 3.1.1 as appropriate.

Days 3-5:

- In Activities 3.1.2 and 3.1.3, students consider implications of Big Data. A writing assignment about these implications is the last step of Activity 3.1.3. This is one of the opportunities students have in CSE to practice the CS Principles Explore Performance Task. Show the last step of Activity 3.1.3 so that students can be alert to possible topics for that task while they complete Activity 3.1.2.
- Introduce Activity 3.1.2 and guide discussion as appropriate.

Days 6-7:

- You will need a paper copy of 3.1.3.a.A Supplement for the class.
- Present BigData.pptx.
- Introduce Activity 3.1.3. Part I of that activity can be completed outside of class. Direct classroom movement during Part II, at which time the supplement is used.
- Present 3.1.3 TheBrain.pptx
- Assign Parts III and IV of the activity and circulate while students use class time to complete these parts or to coordinate worktime outside of class with their partner.

Day 8-10:

- Present or assign 3.1.4 Seeing.pptx and 3.1.4 PieCharts.pptx.
- Introduce Activity 3.1.4 and circulate while students work.
- Assign teams for student work in Problem 3.2.7 (from the next lesson). Provide time for those pairs to establish team norms and brainstorm questions on which they can focus.

Days 11-14:

- Present or assign 3.1.5 Histograms.pptx and 3.1.5 DescribingDistributions.pptx.
- Introduce Activity 3.1.5 and circulate while students complete Parts I, II and III.
- Present 3.1.5 TransformingData and have students complete Parts IV and V of the activity.
- Check in with students about their progress on Problem 3.2.7 (from the next lesson). They should search for data relevant to their focus area and begin to explore relevant data.

Unit 3.2 Discovering Knowledge from Data

The goal of this unit is for students to extend on their knowledge from the previous unit to create a range of visualizations to analyze complex sets of large data, and to meaningfully interpret the patterns they uncover. Students are introduced to statistics and a way to separate coincidental patterns from inference patterns that are scientifically justified. More complicated and non-traditional forms of data are introduced, including image analysis and face recognition, linked data, geographic data, and the data in DNA. Students continue to draw conclusions about themselves from this relevant data, including various geographic perspectives on their life and facial recognition of their own features. The unit lessons use Excel® spreadsheet software as well as the *Python*® programming language to manipulate and visualize data. Students examine geographic and social data using heat maps and directed graphs. Students experiment with object recognition and face recognition. They are challenged to discover clustering and association patterns lurking in data sets distributed across student computers and school sites, such that data cleaning and warehousing are necessary. Finally, student teams choose an essential questions within an area for investigation and make discoveries to answer that question using large data.

Understandings

- U1.02 - Computational artifacts can be evaluated. (S1.06)
- U2.01 - Binary sequences represent digital data. (K2.02, K2.03, K7.08, K2.16)
- U3.01 - Data can be structured to facilitate use. (K3.01) (S3.07)
- U3.03 - Analysis of data can be automated. (K3.03) (S3.05, S3.07, S3.03, S3.08, S5.12)
- U3.04 - Data visualizations are important tools for discovering and communicating knowledge. (S8.06, S3.05, S3.04, S3.03, S3.08)
- U3.05 - The human brain and today's computers have complementary strengths for analyzing data. (S3.04, S3.07, S3.08)
- U3.06 - The size of a data set affects how the data can be used. (K4.04)
- U3.07 - Collecting and managing data raises technical issues regarding storage, access, durability, privacy, and security. (K7.08)
- U3.08 - Ethical and societal issues are raised by the impact of Big Data and require attention from many stakeholders. (K7.09)
- U4.01 - Programs implement algorithms to solve problems. (K2.08, K4.05) (S5.12)
- U4.02 - Algorithms can be analyzed for efficiency, and appropriate algorithms can be selected based upon efficiency. (K4.04) (S4.02)
- U4.03 - Empirical analysis of algorithms requires a systematic approach. (S5.09, S5.13)
- U4.04 - A given algorithmic problem with standard solutions can be applied in diverse contexts. (K2.08)
- U5.02 - Solutions in a programming language are created by breaking a problem apart into component problems. (S5.11)
- U5.03 - Creating solutions with computation requires a persistent, iterative problem-solving approach. (S5.09)
- U5.04 - Programming requires an understanding of mathematical operations and data abstractions. (K3.03, K4.05)
- U5.07 - Programmers create high-level documentation to communicate the purpose and function of their code.
- U5.08 - Programmers must prioritize making their code well-documented and readable for it to be maintained. (S5.09, S5.10)
- U6.03 - The Internet facilitates collaboration. (K7.07)
- U6.05 - The information and processing power on any networked device can be accessed by potentially hostile parties.
- U7.01 - Mobile and networked computing have transformed commerce, social interactions, news sourcing and dissemination, and culture. (K7.07) (S8.07)
- U7.02 - Computing is having profound impacts on individual privacy. (K7.09)
- U7.04 - Making information accessible to all people requires attention from a variety of stakeholders. (K7.07, K7.14)
- U7.08 - Scalability is an important consideration for distributed solutions. (K4.04)
- U8.01 - Computing artifacts and programs can be higher quality as a result of collaboration. (S8.08, S8.06)

- U8.02 - Working in a team requires effective communication, clear responsibilities, and attention to interpersonal relationships. (S8.06)
- U9.01 - Computer science and information technology careers offer creative job opportunities for individuals with a wide variety of backgrounds and goals. (K9.05, K9.04)

Knowledge

- Students will:
- K1.02 - Describe the role of creativity in producing data visualizations.
- K2.02 - Distinguish continuous and discrete phenomena and identify digital and analog data. (U2.01)
- K2.03 - Describe layers of abstraction that help people represent and use data. (U2.01)
- K2.04 - Identify layers of abstraction used in programming languages and describe advantages and disadvantages inherent in working at a high level of abstraction.
- K2.08 - Distinguish deterministic and randomized models and describe the role of Monte Carlo techniques. (U4.01, U4.04)
- K2.16 - Describe the implications of the limited precision of digital information in applications. (U2.01)
- K3.01 - Relate the categorical or quantitative nature of data to the operations and visualizations that are appropriate. (U3.01)
- K3.03 - Describe how mathematical functions and algorithms can be applied to a data set to construct a derived data set. (U3.03, U5.04)
- K3.04 - Identify the relationship between parallel computation and computationally intensive tasks like simulation, modeling, and analysis of large data sets.
- K3.06 - Identify data visualizations as an important tool for communication.
- K4.04 - Estimate the time for an algorithm to operate on a data set, given the algorithm's running time on another data set and the algorithm's complexity in big-O notation. (U3.06, U4.02, U7.08)
- K4.05 - Recognize common patterns employing variables, including value accumulation, list aggregation, and iteration across the elements of a collection. (U4.01, U5.04)
- K7.07 - Analyze access to the Internet and to data among people in different countries, and connect to measures of health and wealth. (U6.03, U7.01, U7.04)
- K7.08 - Discuss societal implications of the persistence and the ease of copying digital information. (U2.01, U3.07)
- K7.09 - Articulate a range of positions on questions related to privacy with respect to Big Data. (U3.08, U7.02)
- K7.14 - Contrast the patterns of inequity characterizing the personal computing revolution and the current mobile computing revolution. (U7.04)
- K9.04 - Describe career-oriented opportunities to use computational skills to positively affect people's lives. (U9.01)
- K9.05 - Identify fields of computing careers. (U9.01)
- K9.06 - Describe how computing is connected to innovations in other fields.
- K9.07 - Describe how specific career fields, e.g., biology and marketing, have changed profoundly as a result of automated collection and processing of data.

Skills

Students will:

- S1.01 - Create a visual artifact of a computational process, such as an image or screenshot of a user interface or web page.
- S1.06 - Analyze a computational artifact for correctness. (U1.02)
- S3.03 - Represent and understand phenomena by identifying relationships among data. (U3.03, U3.04)
- S3.04 - Identify the type of basic visualization appropriate to a particular data set. (U3.04, U3.05)
- S3.05 - Use appropriate tools to construct a scatter plot, histogram, pie chart, or compound bar graph. (U3.03, U3.04)
- S3.07 - Use appropriate tools to create new data sets derived from other data sets. (U3.01, U3.03, U3.05)
- S3.08 - Use appropriate tools to describe a linear relationship (or lack thereof) inferred from data. (U3.03, U3.04, U3.05)
- S4.02 - Evaluate a program for efficiency. (U4.02)
- S5.09 - Improve readability, efficiency, or correctness of code from other people. (U4.03, U5.03, U5.08)
- S5.10 - Document a software product using appropriate in-line comments. (U5.08)
- S5.11 - Encapsulate a set of related statements in a function or procedure. (U5.02)
- S5.12 - Create a function to perform a particular calculation from the function's arguments. (U3.03, U4.01)
- S5.13 - Deduce the value of a variable resulting from execution of particular code. (U4.03)
- S8.06 - Collaborate when collecting and analyzing data to answer a question. (U3.04, U8.01, U8.02)
- S8.07 - Compare to select from among several tools available for collaboration. (U7.01)
- S8.08 - Collaborate when programming. (U8.01)
- S8.11 - Distinguish sources of information and comment on the information's reliability and intended audience.

Essential Questions

- Q1 - How will computation impact fields other than computing itself?
- Q2 - How will computation impact society?
- Q3 - How can patterns be discovered in data?

Day-by-Day Plans

Activity 3.2.1	Inferential Statistics (2 days) (U1.01, U3.01, U3.03, U3.04, U3.05) (K1.02, K2.01, K2.02, K2.03, K2.08, K3.01, K3.03, K3.06) (S2.03, S3.01, S3.02, S3.03, S3.04, S3.05, S3.08, S3.10, S8.06)
Activity 3.2.2	Image Data (1 day) (U7.03, U4.02, U4.01, U4.04, U3.08, U3.07, U3.05, U3.03, U3.02, U5.04, U5.02) (K2.02, K2.03, K2.01, K9.07, K7.09, K3.04) (S2.06)
Activity 3.2.3	Linked Data (1 day)

	(U1.02, U4.01, U7.05, U3.08, U3.07, U3.06, U3.05, U3.04, U3.03) (K1.02, K2.02, K2.03, K2.01, K2.04, K3.06, K3.01, K3.03, K3.02, K2.12, K4.03) (S1.06, S8.06, S3.04, S3.03, S6.04, S3.09)
Activity 3.2.4	Geographic Data (2 days) (U1.02, U7.06, U7.05, U3.08, U3.07, U3.05, U3.04, U3.02, U6.03) (K1.02, K2.02, K2.03, K2.01, K9.07, K9.04, K7.03, K9.02, K3.06, K3.01) (S1.01, S1.06, S1.05, S8.06, S3.04, S3.06, S3.03, S6.04)
Activity 3.2.5	Considering Gattaca (1 day) (U3.08, U7.09, U7.02)
Project 3.2.6	Genomic Data (optional, 3 days) (U7.09)(S7.01) (K7.05, K7.12, K7.13, K9.03, K9.06, K9.07)
Problem 3.2.7	Investigating with Data (6 days) (U1.01, U1.03, U1.02, U4.01, U8.04, U8.05, U3.05, U3.04, U3.03, U9.03, U8.01, U5.04, U5.07, U5.06, U5.01, U5.03, U5.02, U5.08, U8.02, U6.03) (K8.01, K7.12, K3.06, K3.01, K3.03, K4.05) (S1.01, S1.06, S1.05, S8.10, S8.08, S8.14, S8.13, S8.09, S2.06, S8.04, S8.06, S3.05, S3.04, S3.07, S8.05, S3.03, S3.02, S6.04, S5.10, S5.11, S5.12)

Days 1-2:

- Frame the Lesson with Problem 3.2.7 and with the Essential Questions. As an introduction to Activity 3.2.1, ask how students will know if patterns they discover in their work on Problem 3.2.7 are just coincidences. Present 3.2.1.A InferentialStatistics.pptx and circulate while students work on Activity 3.2.1 InferentialStatistics.
- Begin circulating the movie Gattaca if you didn't do so already.

Days 3-4:

- Activities 3.2.2 and 3.2.3 are both short and light hearted. They could be combined in a single day. Introduce them both as an exploration of what can be done with non-traditional types of data.
- Teacher credentials for the Visual Thesaurus at https://pltw.visualthesaurus.com/login/ are email/username pltwteacher and password teacher123.
- Depending on how your students reacted to the P/NP problems, you might connect Activity 3.2.3 to Lesson 2.3 by reminding them of the Traveling Salesperson Problem, coloring a graph so that adjacent nodes don't share a color, or the P=NP? million dollar question.

Days 5-6:

- Introduce Activity 3.2.4 Geographic Data and circulate while students work.

Day 7:

- Lead student discussion about Gattaca using student responses to Activity 3.2.5. If students have not yet completed the activity, this day may be delayed until later in the lesson.

Days 8-10:

- Optionally, use 3 additional days in this lesson to complete Project 3.2.6 Genomic Data.

- If implementing that Project, introduce the Project as a demonstration of the impact of computation in modern biology and in the new methods by which scientists collaborate.
- Refer students to 3.2.6.Pc BiologyVocabulary and give an overview of genetic data using 3.2.6.Pd StandardGeneticCode. The supplement 3.2.6.Pe Alignment and the accompanying source code mutation.py can be used in a format in which pairs collaborate or in which the teacher leads. These three supplements can be used during the potential waiting time during the project, noted below.
- Circulate while students work through Project 3.2.6.
- Step 4d can cause students to wait for 5 to 15 minutes depending on global scientific activity at that time of day. Be prepared to make this time useful using supplements C, D, and E.

Days 8-13:

- At the completion of the lesson, give students appropriate opportunities for sharing the work and for learning from the work of others. Since the deliverables do not include a presentation, a gallery walk will be more effective. Students will be able to see each other's visualizations in a small group format.

Unit 4.1 Moore's Law and Modeling

In this unit, students construct an understanding of how the explosion of technology over the last two decades has impacted every realm of study and employment. Students begin by researching the impact of computer modeling and simulation, which has been made possible by the rapid increase in computational power due to the continued applicability of Moore's Law. They then manipulate discrete electronic components to create logic gates. Students create comparable results using integrated circuits to get a feel for what it means to double the number of transistors that can fit in a given area. Students explore simulation in NetLogo directly by manipulating a model of predation and a model of the spread of viruses in humans. The lesson concludes with an examination of the code of ethics for simulations and reflection on the necessity of adhering to such a code.

Understandings

- U1.02 - Computational artifacts can be evaluated
- U2.01 - Binary sequences represent digital data
- U2.03 - Computing relies on abstractions of hardware represented with software
- U2.07 - Simulation and modeling can help us understand, communicate about, and predict natural phenomena
- U2.08 - Physical systems, like sound or biological molecules, have both digital and analog characteristics
- U3.02 - Our capabilities to collect, store, and process data are changing at profound rates
- U3.03 - Analysis of data can be automated
- U3.04 - Data visualizations are important tools for discovering and communicating knowledge
- U4.01 - Programs implement algorithms to solve problems
- U5.01 - Creating solutions with computation requires exploring the tools available, selecting an appropriate tool, and gaining expertise with the tool

- U5.04 - Programming requires an understanding of mathematical operations and data abstractions
- U7.06 - Crowdsourcing identifies new problems and provides new solutions
- U7.09 - Computing is rapidly and profoundly changing science and engineering
- U9.02 - Parallel computing is a quickly evolving field relevant to hardware, software, and users
- U9.03 - Computational thinking boosts most career paths

Knowledge

Students will...

- K2.02 - Distinguish continuous and discrete phenomena and identify digital and analog data
- K2.05 - Describe any simulation as abstracting some aspects of a model while ignoring other details
- K2.06 - Describe any simulation as relying on assumptions, some of which can be parameterized, with conclusions applicable to a range of parameter values
- K2.07 - Describe an abstraction of hardware
- K2.08 - Distinguish deterministic and randomized models and describe the role of Monte Carlo techniques
- K2.09 - Recognize that chaotic and periodic behavior can be exhibited by both deterministic and randomized models
- K2.11 - Discrete, continuous, and agent-based models using digital computation rely on discrete calculations
- K2.14 - Describe some of the major applications of simulation
- K2.15 - Identify landmarks in hardware development such as tubes, transistors, and VLSI, as well as developments such as optoelectronics, nanotechnology, and quantum computing
- K2.16 - Describe the implications of the limited precision of digital information in applications
- K2.17 - Describe the abstraction of a logic gate and explain how complex logic functions can be constructed from NAND gates
- K3.04 - Identify the relationship between parallel computation and computationally intensive tasks like simulation, modeling, and analysis of large data sets
- K3.06 - Identify data visualizations as an important tool for communication
- K7.03 - Describe examples of how location-aware computing is transforming infrastructure
- K7.10 - Describe examples of crowdsourcing being used for large data set collection or collective biological processing power
- K7.11 - Describe examples in which computation has or will create new societal phenomena and human capabilities to perceive and act upon our environment

Skills

Students will...

- S1.06 - Analyze a computational artifact for correctness
- S2.01 - Identify and explain some of the assumptions made by a particular model

- S2.03 - Identify quantities relevant to a phenomenon and explain the implications of abstracting to a particular model that accounts for some parameters and ignores others
- S2.04 - Manipulate the parameters of a simulation to identify how the parameters affect the behavior being modeled
- S2.08 - Compare hardware and software manipulations that produce similar outcomes
- S2.09 - Construct a circuit including discrete components and semiconductor chips from a symbolic diagram
- S3.01 - Interpret data and data visualizations to reach conclusions about a model's behavior
- S3.03 - Represent and understand phenomena by identifying relationships among data
- S5.06 - Evaluate programs written by others for readability
- S5.08 - Identify appropriate boundary conditions for testing a program
- S8.04 - Effectively manage a project, including planning and time management, team norming and load balancing, file/revision management, and documentation
- S8.10 - Present original research and research from literature
- S8.11 - Distinguish sources of information and comment on the information's reliability and intended audience

Essential Questions

- How are simulations created from models?
- How are simulation models similar to and different from reality?
- How are modeling and simulation impacting other creative fields?

Day-by-Day Plans

Activity 4.1.1	Computing Impacts All Fields (2 days) *U3.02, U7.06, U7.09, U9.02, U9.03, S8.04, S8.10, S8.11, K2.14, K3.04, K7.03, K7.10, K7.11*
Activity 4.1.2	Basic Control Circuits (3 days) *U2.01, U2.03, U5.04, S2.08, S2.09, K2.07, K2.15, K2.17*
Activity 4.1.3	Introducing Simulations (3 days) *U2.07, U2.08, U3.03, U3.04, U4.01, U5.01, K2.02, K2.05, K2.06, K2.08, K2.09, K2.11, K2.16, K3.06, S1.06, S2.01, S2.03, S2.04, S3.01, S3.03, S5.06, S5.08*
Activity 4.1.4	Varying Parameters (2 days) *U1.02, U2.07, U3.04, K3.06*
Activity 4.1.5	Assumptions, Abstractions, and Ethics (2 days) *U2.08, S1.06, S2.01, S2.03, S5.06, K2.02, K2.05, K2.06, K2.08, K2.11, K2.16*

Days 1 – 2:

- Preview the lesson with students, framing the whole lesson as being tied together with the applicability of Moore's law and the impact it has had on all fields and careers, emphasizing simulation and modeling as becoming explosively more useful as processing power increases.

- Guide students through the 4.1.1 presentation.
- Guide students through the remainder of Activity 4.1.1.
- Facilitate student research and presentation of research in Part III.
- Assign conclusion questions.

Days 3 – 5:

- Review conclusion questions from 4.1.1.
- Frame this activity for students as an opportunity to explore what happens at the lowest level of abstraction within a computer.
- Guide students through the 4.1.2 presentation.
- Distribute electronic breadboarding kits.
- Guide students through Activity 4.1.2.
- Assign conclusion questions.

Days 6 – 8:

- Review conclusion questions from 4.1.2.
- Introduce NetLogo as a tool for simulation and modeling that will allow students to quickly observe, interact with, and analyze new models of phenomena from a variety of fields.
- Guide students through the 4.1.3 slideshow detailing the basic interface components of NetLogo.
- Guide students through Activity 4.1.3.
- Assign conclusion questions.

Days 9 – 10:

- Review conclusion questions from 4.1.3.
- Guide students through Activity 4.1.4. Be prepared to help them with the BehaviorSpace interface.
- Assign conclusion questions.

Days 11 – 12:

- Review conclusion questions from 4.1.4.
- Guide students through the slideshow 4.1.5 Assumptions.
- Allow students some time to experiment independently with the Virus model identified in Step 4 of Activity 4.1.5.
- Guide students through the rest of Part I of Activity 4.1.5.
- Be prepared to clarify as needed regarding the simulationist code of ethics in Part II of Activity 4.1.5.
- Assign conclusion questions.

Lesson 4.2 Intelligent Agents

In this lesson students experiment with materials designed to illuminate the rise of intelligent and complex behavior from simple rules and seemingly unintelligent agents. Students begin by studying a model of Langton's ant, a simple Turing machine with some surprising emergent behavior. The students manipulate artificial neurons, called perceptrons, and neural networks. Students design and conduct their own experiments on a model of their own choosing using Monte Carlo methods. Students explore the generation and observation of fractals and study a diffusion-limited aggregation model for producing fractal behavior. In the final project of the course, students choose a tool or tools that they have learned about in the course and apply their knowledge to create a novel product of their own design. They present their product to their class along with reflections about how it is tied to everything they've learned about Computer Science.

Understandings

- U1.01 - Computing fosters creative expression, sometimes resulting in artifacts (K1.02) (S1.01)
- U1.02 - Computational artifacts can be evaluated (S1.06)
- U1.03 - Programming is a creative endeavor (S5.03)
- U2.01 - Binary sequences represent digital data
- U2.02 - Computing relies on layers of abstraction in software (K2.04, K2.05, K2.16) (S5.12)
- U2.04 - Abstraction allows for simple utilization of other people's code
- U2.06 - The solution to one problem can be applied to another seemingly unrelated problem by identifying and reusing a pattern (K2.10) (S5.03)
- U2.07 - Simulation and modeling can help us understand, communicate about, and predict natural phenomena (K2.08, K2.09, K2.06, K2.13, K2.14) (S2.02, S2.03, S2.01, S2.04, S3.01)
- U2.09 - Intelligent behavior emerges from networked collections of simple algorithms (K2.13, K2.14)
- U3.01 - Data can be structured to facilitate use
- U3.03 - Analysis of data can be automated
- U3.04 - Data visualizations are important tools for discovering and communicating knowledge
- U3.05 - The human brain and today's computers have complementary strengths for analyzing data
- U4.01 - Programs implement algorithms to solve problems (K2.08, K5.02) (S5.12)
- U4.03 - Empirical analysis of algorithms requires a systematic approach (S5.02)
- U4.04 - A given algorithmic problem with standard solutions can be applied in diverse contexts
- U5.01 - Creating solutions with computation requires exploring the tools available, selecting an appropriate tool, and gaining expertise with the tool
- U5.03 - Creating solutions with computation requires a persistent, iterative problem-solving approach
- U5.04 - Programming requires an understanding of mathematical operations and data abstractions (K5.02, K3.03)

- U5.05 - The user interface of a piece of software can greatly affect how it is used
- U5.08 - Programmers must prioritize making their code well-documented and readable for it to be maintained (S5.10)
- U7.09 - Computing is rapidly and profoundly changing science and engineering (K9.04, K2.13, K2.14)
- U8.01 - Computing artifacts and programs can be higher quality as a result of collaboration (S8.08, S8.09, S8.06)
- U8.02 - Working in a team requires effective communication, clear responsibilities, and attention to interpersonal relationships (S8.09, S8.06)
- U8.04 - Creative ideas and technical solutions must be communicated in a clear and concise manner
- U8.05 - How people present themselves affects how their work is received
- U9.01 - Computer science and information technology careers offer creative job opportunities for individuals with a wide variety of backgrounds and goals
- U9.03 - Computational thinking boosts most career paths (K9.08, K9.07, K9.04, K9.06, K3.04, K2.14)

Knowledge

Students will...

- K2.05 - Describe any simulation as abstracting some aspects of a model while ignoring other details (U2.02)
- K2.06 - Describe any simulation as relying on assumptions, some of which can be parameterized, with conclusions applicable to a range of parameter values (U2.07, U2.08)
- K2.08 - Distinguish deterministic and randomized models and describe the role of Monte Carlo techniques (U2.07, U4.01)
- K2.09 - Recognize that chaotic and periodic behavior can be exhibited by both deterministic and randomized models (U2.07)
- K2.13 - Give examples of synergetic properties that can emerge from real or simulated collections of independent agents (U2.07, U2.09, U7.09)
- K2.14 - Describe some of the major applications of simulation (U2.07, U2.09, U7.09, U9.03)
- K3.03 - Describe how mathematical functions and algorithms can be applied to a data set to construct a derived data set (U5.04)
- K3.06 - Identify data visualizations as an important tool for communication
- K4. 01 - Describe a computer as responding to input in a deterministic manner that depends only on input and on the computer's state
- K5. 03 - Distinguish among a variety of educational and reference resources related to code libraries
- K8.01 - Describe a version control system
- K9.03 - Describe the impact of genomic data and computation in medicine, biology, and anthropology

Skills

Students will...

- S1.01 - Create a visual artifact of a computational process, such as an image or screenshot of a user interface or web page (U1.01)
- S1.06 - Analyze a computational artifact for correctness (U1.02)
- S2.01 - Identify and explain some of the assumptions made by a particular model (U2.07)
- S2.02 - Propose a method for simulating a natural phenomenon of interest (U2.07, U5.02)
- S2.03 - Identify quantities relevant to a phenomenon and explain the implications of abstracting to a particular model that accounts for some parameters and ignores others (U2.07, U2.08)
- S2.04 - Manipulate the parameters of a simulation to identify how the parameters affect the behavior being modeled (U2.07)
- S3.01 - Interpret data and data visualizations to reach conclusions about a model's behavior (U2.07)
- S3.03 - Represent and understand phenomena by identifying relationships among data
- S3.04 - Identify the appropriate type of basic visualization appropriate to a particular data set
- S3.08 - Use appropriate tools to describe a linear relationship (or lack thereof) inferred from data
- S5.02 - Analyze and test code from other people (U4.03)
- S8.01 - Document a software development process
- S8.02 - Document a software product using high-level documentation
- S8.03 - Use a version control system
- S8.04 - Effectively manage a project, including planning and time management, team norming and load balancing, file/revision management, and documentation
- S8.05 - Collaborate effectively with others when managing a project
- S8.06 - Collaborate when collecting and analyzing data to answer a question (U3.02, U8.01, U8.02)
- S8.09 - Collaborate when presenting (U8.01, U8.02)
- S8.13 - Communicate the progress on a project, including accomplishments and next steps
- S8.14 - Present a plan for producing knowledge

Essential Questions

- ➢ How has computation affected our ability to predict the future?
- ➢ How has computation affected our ability to experience virtual phenomena?
- ➢ How has simulation changed the process of scientific investigation and knowledge generation?

Day-by-Day Plans

Activity 4.2.1 Emergent Behavior (2 days)

U1.02, U2.01, U2.06, U2.07, U2.09, U3.01, U3.04, U4.01, U4.04, S2.04, S5.02, S8.06, K2.08. K2.09, K2.13, K4.01, K9.03

Activity 4.2.2 Neural Networks (3 days)

U1.02, U2.01, U2.07, U2.09, U4.03, U5.04, S1.06, S2.01, S2.03, S2.04, K2.05, K2.06, K3.03, K4.01

Project 4.2.3 Modifying a Simulation's Assumptions (5 days)
U1.02, U2.02, U2.04, U2.07, U3.04, U3.05, U5.03, U5.05, U8.01, U8.02, U8.04, U8.05, S1.01, S1.06, S2.01, S2.02, S2.03, S2.04, S3.01, S3.03, S3.04, S8.01, S8.02, S8.03, S8.04, S8.05, S8.06, S8.09, S8.13, S8.14, K2.05, K2.06,K2.08, K2.13, K3.03, K3.06, K5.03, K8.01

Activity 4.2.4 Beauty in Chaos and Fractals (2 days)
U1.01, U1.03, U2.07, U2.09, U7.09, S1.01, S2.01, S2.03, S2.04, K2.05, K2.06, K2.13, K2.14, K3.03

Project 4.2.5 Computer Science Principles (10 day)
U1.01, U2.02, U2.04, U3.01, U3.03, U3.04, U4.01, U4.03, U4.04, U5.01, U5.04, U5.08, U7.09, U8.01, U8.05, U9.01, U9.03, S8.04, S8.09

Time: 22 days

Day 1 – 2:

- Preview the lesson with students, framing the whole lesson as being a final foray into NetLogo before students choose their own independent CS projects to work on until the end of the course. Share the details of the final presentation with them.
- Guide students through the 4.2.1 slideshow, which provides a brief introduction to NEtLogo code
- Guide students through the remainder of Activity 4.2.1.
- In Step 21 students brainstorm as a class. In Step 22 they conduct their own experiments and submit data visualizations.
- Assign conclusion questions.

Days 3 – 5:

- Review conclusion questions from 4.2.1
- Guide students through the 4.2.2 slideshow on biological neurons. They will examine artificial neurons in the activity.
- Guide students through Activity 4.2.2.
- Assign conclusion questions.

Days 6 – 10:

- Review conclusion questions from 4.2.2.
- Students will choose a model and modify the assumptions in that simulation. In some cases this can be done by simply using the UI. In others it will require writing code. Guide students toward appropriate tasks for their readiness level: The wolf sheep predation model and the page rank model can both be modified through the UI.
- Guide students through Activity 4.2.3.
- Assign conclusion questions.

Days 11 – 12:

- Review conclusion questions from 4.2.3.
- Guide students through Activity 4.2.4, and be prepared assist with the XaoS interface.
- Assign conclusion questions.

Day 13 – 22:

- Review conclusion questions from 4.2.4.
- This Project is the student's chance to take something they wish they'd had more time to work on and polish it. Students may also choose to pick up a new project using the tools and methods they've explored in the course. It is a good opportunity for students to feel like they've gained mastery.
- Dedicate time in the last couple classes for student presentations so that students can show off their work.