PIANO

BY WONGCHANT

CURRICULUM OUTLINE COMPUTER SCIENCE DISCOVERIES

A Guide for Computer Science Discoveries Curriculum

COMPUTER SCIENCE DISCOVERIES

10 - 15 YEARS

MODULE 1

UNIT 1

PROBLEM SOLVING AND COMPUTING

DESCRIPTION

Problem Solving and Computing is a highly interactive and collaborative introduction to the field of computer science, as framed within the broader pursuit of solving problems. You'll practice using a problem solving process to address a series of puzzles, challenges, and real world scenarios. Next, you'll learn how computers input, output, store, and process information to help humans solve problems. The unit concludes with a project in which you design an application that helps solve a problem of your choosing.

Implementation Guidance for Problem Solving and Computing
The first chapter of this unit should be completed before any other unit in CS Discoveries
Alternate lessons are provided for some lessons in this unit, depending on your classroom context
You can view the full Curriculum Guide and Implementation Guide for more information about this unit

- ➢ Survey
  - ○ CS Discoveries Pre-survey

- ➢ **Lesson 1: Intro to Problem Solving**
  - ○ **DESCRIPTION:** In this lesson, students work in groups to design aluminum foil boats that will support as many pennies as possible. Groups have two rounds to work on their boats, with the goal of trying to hold more pennies than they did in round 1. The structure of the activity foreshadows different steps of the problem solving process that students will be introduced to in more detail in the following lesson. At the end of the lesson students reflect on their experiences with the activity and make connections to the types of problem solving they will be doing for the rest of the course.
  - ○ Alternate versions of this lesson are also available.
    - ■ Newspaper Table
    - ■ Spaghetti Bridge
    - ■ Paper Tower

- ➢ **Lesson 2: The Problem Solving Process**
  - ○ **DESCRIPTION:** This lesson introduces the formal problem solving process that students will use over the course of the year, Define - Prepare - Try - Reflect. The lesson begins by asking students to brainstorm all the different types of problems that they encounter in everyday life. Students are then shown the four steps of the problem solving process and work together to relate these abstract steps to their actual experiences solving problems. First students relate these steps to the problem activities from the previous lesson, then a problem they are good at solving, then a problem they want to improve at solving. At the end of the lesson the class collects a list of generally useful strategies for each step of the process to put on posters that will be used throughout the unit and year.
    - ■ Video: Problem Solving Process

MODULE 1

- **Lesson 3: Exploring Problem Solving**
  - **DESCRIPTION:** In this lesson students apply the problem solving process to three different problems in order to better understand the value of each step. They will solve a word search, arrange seating for a birthday party, and redesign a classroom. The problems grow increasingly complex and poorly defined to highlight how the problem solving process is particularly helpful when tackling these types of problems. The lesson concludes with students reflecting on their experience with the problem solving process. They will justify the inclusion of each step and will brainstorm questions or strategies that can help them better define open-ended problems, as this is often the most critical step.
    This lesson will likely take two class periods or more to complete. The first two problems may fit into a single class period but the third will need to be moved to a second day.
  - Alternate versions of this lesson are also available.
    - Animal Theme
    - Games Theme

- **Lesson 4: What is a Computer?**
  - **DESCRIPTION:** In this lesson students develop a preliminary definition of a computer. To begin the lesson, the class will brainstorm possible definitions for a computer and place the results of this brainstorm on the board. Next, students will work in groups to sort pictures into “is a computer” or “is not a computer” on poster paper. Groups will place their posters around the room and briefly explain their motivations for choosing some of their most difficult categorizations. The teacher will then introduce a definition of the computer and allow students to revise their posters according to the new definition.
    - Video: What Makes a Computer, a Computer?

- **Lesson 5: Input and Output**
  - **DESCRIPTION:** In this lesson students consider how computers get and give information to the user through inputs and outputs. Students first consider what information they would need to solve a "thinking problem", then use that information to produce a recommendation. They then identify the inputs and outputs to that process. Afterwards, students consider an app that engages in the same process and determine how that app inputs and outputs information. Last, they consider other types of inputs and outputs that computers can use to help solve problems.
    - App: Pet Chooser
    - App: Story Creator
    - App: Improved Pet App

- **Lesson 6: Processing**
  - **DESCRIPTION:** This lesson introduces students to four common types of processing: if/then (conditionals), finding a match (searching), counting, and comparing. Students are first introduced to the types of processing through several sample apps. They then investigate more apps to determine what sorts of processing each uses. They then think of their own app and decide what types of processing it would need to work. Finally, they brainstorm other types of processing that may be useful but were not included in the main lesson.
    - App: Is It Your Birthday?
    - App: National Parks
    - App: How Many Countries
    - App: My Famous Birthday
    - App: Stamp Notebook
    - App: The Fastest Finger
    - App: Guess the Number
    - App: Where Should I Live?

- **Lesson 8: Project - Propose an App**
  - **DESCRIPTION:** To conclude their study of the problem solving process and the input/output/store/process model of a computer, students will propose an app designed to solve a real world problem. This project will be completed across multiple days and will result in students creating a poster highlighting the features of their app that they will present to their classmates. A project guide provides step by step instructions for students and helps them organize their thoughts. The project is designed to be completed in pairs though it can be completed individually.
    - End-of-Unit Survey

- **Lesson 9: Intro to Problem Solving - Newspaper Table (Alternate Lesson 1)**
  - **DESCRIPTION:** In this lesson, students work in groups to design newspaper tables that will hold as many books as possible. Groups have two rounds to work on their tables, with the goal of trying to hold more books than they did in the first round. The structure of the activity foreshadows different steps of the problem solving process that students will be introduced to in more detail in the following lesson. At the end of the lesson students reflect on their experiences with the activity and make connections to the types of problem solving they will be doing for the rest of the course.

    This is an alternate activity to Intro to Problem Solving - Aluminum Boats

- **Post-Project Test**

- **Lesson 10: Intro to Problem Solving - Spaghetti Bridge (Alternate Lesson 1)**
  - **DESCRIPTION:** In this lesson, students work in groups to design spaghetti bridges that will support as many books as possible. Groups have two rounds to work on their bridges, with the goal of trying to hold more books than they did in Round 1. The structure of the activity foreshadows different steps of the problem solving process that students will be introduced to in more detail in the following lesson. At the end of the lesson students reflect on their experiences with the activity and make connections to the types of problem solving they will be doing for the rest of the course.

    This is an alternate activity to Intro to Problem Solving - Aluminum Boats

MODULE 1

- **Lesson 11: Intro to Problem Solving - Paper Tower (Alternate Lesson 1)**
  - **DESCRIPTION:** In this lesson, students work in groups to design paper towers that can stand as high as possible. Groups have two rounds to work on their towers, with the goal of trying to go higher than they did in Round 1. The structure of the activity foreshadows different steps of the problem solving process that students will be introduced to in more detail in the following lesson. At the end of the lesson students reflect on their experiences with the activity and make connections to the types of problem solving they will be doing for the rest of the course.

    This is an alternate activity to Intro to Problem Solving - Aluminum Boats

- **Lesson 13: Exploring Problem Solving - Games Theme (Alternate Lesson 3)**
  - **DESCRIPTION:** In this lesson students apply the problem solving process to three different problems in order to better understand the value of each step. They will solve a maze, organizing a team to race as fast as possible, and design a game. The problems grow increasingly complex and poorly defined to highlight how the problem solving process is particularly helpful when tackling these types of problems. The lesson concludes with students reflecting on their experience with the problem solving process. They will justify the inclusion of each step and will brainstorm questions or strategies that can help them better define open-ended problems, as this is often the most critical step.

    This lesson will likely take two class periods or more to complete. The first two problems may fit into a single class period but the third will need to be moved to a second day.

    This is an alternate activity to Exploring Problem Solving

- **Lesson 12: Exploring Problem Solving - Animals Theme (Alternate Lesson 3)**
  - **DESCRIPTION:** In this lesson students apply the problem solving process to three different problems in order to better understand the value of each step. They will solve tangrams, choose a pet for several people, and plan a pet adoption event. The problems grow increasingly complex and poorly defined to highlight how the problem solving process is particularly helpful when tackling these types of problems. The lesson concludes with students reflecting on their experience with the problem solving process. They will justify the inclusion of each step and will brainstorm questions or strategies that can help them better define open-ended problems, as this is often the most critical step.

    This lesson will likely take two class periods or more to complete. The first two problems may fit into a single class period but the third will need to be moved to a second day.

    This is an alternate activity to Exploring Problem Solving

UNIT 2

WEB DEVELOPMENT

DESCRIPTION

In Web Development, students are empowered to create and share content on their own web pages. They begin by thinking about the role of the web and how it can be used as a medium for creative expression. As students develop their pages and begin to see themselves as programmers, they are encouraged to think critically about the impact of sharing information online and how to be more critical consumers of content. They are also introduced to problem solving as it relates to programming while they learn valuable skills such as debugging, using resources, and teamwork. At the conclusion of the unit, students will have created a personal website they can publish and share.

Implementation Guidance for Web Development

Complete the first three lessons of the Problem Solving and Computing unit before starting

Guidance for how to support students in programming levels and differentiate tasks are available in the Programming Levels Guide and Differentiation Guide

You can view the full Curriculum Guide and Implementation Guide for more information about this unit

- ➢ **Lesson 1: Exploring Web Pages**
  - ○ **DESCRIPTION:** Every website has a purpose, a reason someone created it and others use it. In this lesson, students will start to consider the purposes a website might serve, both for the users and the creators. Students will explore some sample websites and discuss the purpose they serve for their creators. They are then asked to reflect on reasons that someone might want to create websites.
    - ■ Sample Personal Web Pages

- ➢ **Lesson 2: Intro to HTML**
  - ○ **DESCRIPTION:** In this lesson students are introduced to HTML as a solution to the problem of how to communicate both the content and structure of a website to a computer. The lesson begins with a brief unplugged activity demonstrating the challenges of effectively communicating the structure of a web page. Students then look at an exemplar HTML page in Web Lab and discuss with their classmates how HTML tags help solve this problem. Students then write their first HTML. A wrap-up discussion helps to solidify the understanding of content vs. structure that was developed throughout the lesson.
    - ■ Experiment with Web Lab
    - ■ Video: Intro to Web Lab - Part 1
    - ■ Explore HTML
    - ■ Video: Intro to Web Lab - Part 2
    - ■ Practice
    - ■ Assessment
    - ■ Challenge

MODULE 1

- **Lesson 3: Headings**
  - **DESCRIPTION:** In this lesson, students continue to use HTML to structure text on web pages, this time in pairs, with a focus on working together and debugging problems with their sites. Students learn how the different heading elements are displayed by default and practice using them to create page and section titles.
    - Video: Pair Programming
    - Skill Building
    - Quick Check
    - Video: Debugging
    - Practice
    - Assessment
    - Challenges

- **Lesson 4: Mini-Project: HTML Web Page**
  - **DESCRIPTION:** In this optional mini-project, students use what they have learned to create their own personal web page on a topic of their choice. The lesson starts with a review of the HTML that students have learned. They then begin their project by designing a web page and identifying which tags they will use to implement it. They then create their web pages in Web Lab and share with the class. Optionally, after engaging in a formal feedback process, they may make final changes to their websites before reflecting on their process. This project can be completed in one day or expanded over several days, depending on the scheduling needs of the class.
    - Review: HTML
    - Challenge: Lists
    - Code Your Page
    - Review Your Page

- **Lesson 5: Digital Footprint**
  - **DESCRIPTION:** This lesson takes a step back from students' work developing web pages to help them articulate what personal information they choose to share digitally and with whom. It also reinforces the notion that much of the information that they choose to share digitally falls largely out of their control the moment it is released.

    Students look at several social media pages to determine what sorts of information people are sharing about themselves or one another. Last, students reflect on what guidelines they think are appropriate for posting information online.

    The ultimate point of this lesson is not to scare students, but rather to experientially bring students to realize that they don't have control over information released online.
    - Social Sleuth

- **Lesson 7: Mini-Project: Your Personal Style**
  - **DESCRIPTION:** In this optional mini-project, students use what they have learned to create their own styled web page on a topic of their choice. The lesson starts with a review of the CSS that students have learned. They then begin their project by designing a web page and identifying which properties they will use to implement it. They create their web pages in Web Lab and share with the class. After engaging in a formal feedback process, they make final changes to their websites before reflecting on their process. This project can be completed in one day or expanded over several days, depending on the scheduling needs of the class.
    - Review: CSS
    - Challenge: RGB Colors
    - Add Content and Structure to Your Page
    - Add Style to Your Page
    - Review Your Page

MODULE 1

- **Lesson 8: Intellectual Property**
  - **DESCRIPTION:** Starting with a discussion of their personal opinions on how others should be allowed to use their work, the class explores the purpose and role of copyright for both creators and users of creative content. They then move on to an activity exploring the various Creative Commons licenses as a solution to the difficulty in dealing with copyright.

- **Lesson 9: Using Images**
  - **DESCRIPTION:** Students start the class by considering the ethical implications of using images on their websites, specifically in terms of intellectual property. They then learn how to add images to their web pages using the <img> tag and how to cite the image sources appropriately.
    - Skill Building
    - Practice
    - Assessment
    - Challenges

- **Lesson 10: Websites for Expression**
  - **DESCRIPTION:** In this lesson students investigate ways to use websites as a means of personal expression and develop a list of topics and interests that they would want to include on a personal website. Students begin by brainstorming different ways that people express and share their interests and ideas. Students then look at a few exemplar websites to identify ways they express ideas. Finally, students brainstorm and share a list of topics and interests they might want to include on a personal website; they will reference this list as they progress through the unit.
    - Sample Personal Websites

- **Lesson 11: Styling Elements with CSS**
  - **DESCRIPTION:** This lesson continues the introduction to CSS style properties, this time focusing more on non-text elements. Students begin this lesson by looking at a website about Desserts of the World. They investigate and modify the new CSS styles on this website, adding their own styles to the page. After working on the Desserts page, students apply their knowledge of new CSS properties to their personal websites.
    - Sample Website
    - Exploration
    - Skill Building
    - Practice
    - Assessment
    - Challenges

- **Lesson 12: Your Web Page - Prepare**
  - **DESCRIPTION:** In this lesson, students engage in the "prepare" stage of the problem-solving process, deciding what elements and style their web page will have. They begin by reviewing the different HTML, CSS, and digital citizenship guidelines they will need in building their web pages. They then describe and sketch their pages, listing the tags and styles they will use to get the layout and design that they decided on. They then move online to find and download the images they will need for their pages. Afterwards, they reflect on how their plan will ensure that the website does what it is designed to do.

- **Lesson 13: Project - Personal Web Page**
  - **DESCRIPTION:** After quickly reviewing their debugging process, students go online to create the pages that they have planned out in previous lessons, with the project guides as a reference. Once they have finished with their pages, they complete a short reflection on the process, what they are most proud of, and what they would like feedback on. They then engage in a structured peer feedback process before making final edits to the pages. Afterwards, they reflect on the skills and practices that helped them to be successful.
    - Project Guide
    - Website Project
    - Video: Publishing Your Website
    - Publishing Your Website
    - Reflection

- **Lesson 14: Websites for a Purpose**
  - **DESCRIPTION:** In this lesson, students explore the different reasons people make websites. They first think of different reasons that they visit websites, then investigate web sites that have been created to address a particular problem. After deciding what different purposes those websites might serve for the creators, they begin to think about the problem that they might want to solve with a web site. At the end of the lesson, students form the groups that they will be in for their chapter projects.
    - Top Websites

- **Lesson 15: Team Problem Solving**
  - **DESCRIPTION:** Students work together to set group norms and brainstorm what features they would like their websites to have. The class starts by thinking of some popular teams in different contexts, then reflects on what makes teams successful. They then get into their own teams and make a plan for how they will interact and reach success in their own projects. Afterwards, the teams begin to brainstorm ideas for their website project.
    - Video: Teamwork
    - Video: Managing Disagreement

MODULE 1

- ➢ **Lesson 16: Sources and Research**
  - **DESCRIPTION:** This lesson encourages students to think more about how to find relevant and trustworthy information online. After viewing and discussing a video about how search engines work, students will search for information relevant to their site. They'll need to analyze the sites they find for credibility to decide which are appropriate to use on their own website.
    - Video: How Search Works

- ➢ **Lesson 17: CSS Classes**
  - **DESCRIPTION:** This lesson expands on the CSS that students have already learned by introducing classes, which allow web developers to treat groups of elements they want styled differently than other elements of the same type. Students first investigate and modify classes on various pages, then create their own classes and use them to better control the appearance of their pages. They then reflect on how they could use this skill to improve their team websites.
    - Sample Web Page
    - Exploration
    - Skill Building
    - Practice
    - Assessment
    - Challenges

- ➢ **Lesson 18: Planning a Multi-Page Site**
  - **DESCRIPTION:** Students work in teams to plan out their web sites and create a sketch of each page. They then download the media that they will need for their sites. At the end of the activity, they decide how the work will be distributed among team members and report whether the entire group agreed to the plan.
    - Upload Your Images

- ➢ **Lesson 19: Linking Pages**
  - **DESCRIPTION:** Students begin the lesson by looking online for the first web page and discussing how its use of links was what started the web. They then transition to Web Lab where they learn how to make their own links, as well as good conventions that make it easier for users to navigate on a page. Finally, they reflect on their group project and what their personal goals are for the final stretch.
    - Exploration
    - Skill Building
    - Practice
    - Adding Pages
    - Challenges

➢ **Lesson 20: Project - Website for a Purpose**

- **DESCRIPTION:** Teams have spent a lot of time throughout the chapter planning their websites. In this lesson they are finally able to code their pages. Using the project guide, the team works together and individually to code all of the pages, then puts all of the work together into a single site.
  - Your Team Project
  - Add Content and HTML
  - Share Your Pages
  - Add Style
  - Share Your Stylesheets
  - Check Your Work

➢ **Post-Project Test**

➢ **Lesson 21: Peer Review and Final Touches**

- **DESCRIPTION:** This lesson focuses on the value of peer feedback. Students first reflect on what they are proud of, and what they would like feedback on. Teams then work with peers to get that feedback through a structured process that includes the project rubric criteria. Afterwards, students decide how they would like to respond to the feedback and put the finishing touches on their sites. After a final review of the rubric, they reflect on their process. To cap off the unit, they will share their projects and also a overview of the process they took to get to that final design.
  - Incorporate Feedback
  - End-Of-Unit Survey

➢ **Lesson 22: CS Discoveries Post-Course Survey**

- Post-Course Survey Instructions

UNIT 3

INTERACTIVE ANIMATIONS AND GAMES

DESCRIPTION

In the Interactive Animations and Games unit, students create programmatic images, animations, interactive art, and games. Starting off with simple, primitive shapes and building up to more sophisticated sprite-based games, students become familiar with the programming concepts and the design process computer scientists use daily. They then learn how these simpler constructs can be combined to create more complex programs. In the final project, students develop a personalized, interactive program.

Implementation Guidance for Interactive Animations and Games

Complete the first three lessons of the Problem Solving and Computing unit before starting

Guidance for how to support students in programming levels and differentiate tasks are available in the Programming Levels Guide and Differentiation Guide

You can view the full Curriculum Guide and Implementation Guide for more information about this unit

- **Lesson 1: Programming for Entertainment**
  - **DESCRIPTION:** Students are asked to consider the "problems" of boredom and self expression, and to reflect on how they approach those problems in their own lives. From there, students will explore how computer science in general, and programming specifically, plays a role in either a specific form of entertainment or as a vehicle for self expression.
    - Sample Programs

- **Lesson 2: Plotting Shapes**
  - **DESCRIPTION:** Students explore the challenges of communicating how to draw with shapes and use a tool that introduces how this problem is approached in Game Lab. The warm up activity quickly demonstrates the challenges of communicating position without some shared reference point. In the main activity, students explore a Game Lab tool that allows students to interactively place shapes on Game Lab's 400 by 400 grid. They then take turns instructing a partner how to draw a hidden image using this tool, accounting for many challenges students will encounter when programming in Game Lab. Students optionally create their own image to communicate before a debrief discussion.
    - Drawing Shapes

MODULE 1

Lesson 3: Drawing in Game Lab

- **DESCRIPTION:** Students are introduced to Game Lab, the programming environment for this unit, and begin to use it to position shapes on the screen. They learn the basics of sequencing and debugging, as well as a few simple commands. At the end of the lesson, students will be able to program images like the ones they made with the drawing tool in the previous lesson.

  - Introduction to Game Lab
  - Video: Drawing in Game Lab - Part 1
  - Using the Grid
  - Video: Drawing in Game Lab - Part 2
  - Skill Building
  - Practice
  - Assessment
  - Challenges

Lesson 4: Shapes and Parameters

- **DESCRIPTION:** In this lesson students continue to develop their familiarity with Game Lab by manipulating the width and height of the shapes they use to draw. The lesson kicks off with a discussion that connects expanded block functionality (e.g. different sized shapes) with the need for more block inputs, or "parameters". Students learn to draw with versions of `ellipse()` and `rect()` that include width and height parameters. They also learn to use the `background()` block.

  - Sample Web Page
  - Exploration
  - Skill Building
  - Practice
  - Assessment
  - Challenges

Lesson 5: Variables

- **DESCRIPTION:** In this lesson students learn how to use variables to label a number. Students begin the lesson with a very basic description of the purpose of a variable within the context of the storage component of the input-output-storage-processing model. Students then complete a level progression that reinforces the model of a variable as a way to label or name a number.

  - Prediction
  - Video: Introduction to Variables
  - Skill Building
  - Practice
  - Assessment
  - Challenges

Lesson 6: Random Numbers

- **DESCRIPTION:** Students are introduced to the `randomNumber()` block and how it can be used to create new behaviors in their programs. They then learn how to update variables during a program. Combining all of these skills, students draw randomized images.

  - Exploration
  - Skill Building
  - Practice
  - Assessment
  - Challenges

MODULE 1

- **Lesson 7: Sprites**
  - **DESCRIPTION:** In order to create more interesting and detailed images, students are introduced to the sprite object. The lesson starts with a discussion of the various information that programs must keep track of, then presents sprites as a way to keep track of that information. Students then learn how to assign each sprite an image, which will greatly increase the complexity of what they can draw on the screen.
    - Exploration
    - Video: Introduction to Sprites
    - Skill Building
    - Video: The Animation Tab
    - Skill Building
    - Practice
    - Assessment
    - Challenges

- **Lesson 8: Sprite Properties**
  - **DESCRIPTION:** Students extend their understanding of sprites by interacting with sprite properties. Students start with a review of what a sprite is, then move on to Game Lab to practice more with sprites, using their properties to change their appearance. They then reflect on the connections between properties and variables.
    - Prediction
    - Skill Building
    - Practice
    - Assessment
    - Challenges

- **Lesson 9: Text**
  - **DESCRIPTION:** This lesson introduces Game Lab's text commands, giving students more practice using the coordinate plane and parameters. At the beginning of the lesson, students are asked to caption a cartoon created in Game Lab. They then move onto Code Studio where they practice placing text on the screen and controlling other text properties, such as size. Students who complete the assessment early may go on to learn more challenging blocks related to text properties.
    - Prediction
    - Skill Building
    - Practice
    - Assessment
    - Challenges

- **Lesson 10: Mini-Project - Captioned Scenes**
  - **DESCRIPTION:** After a quick review of the code they have learned so far, students are introduced to their first creative project of the unit. Using the problem-solving process as a model, students define the scene that they want to create, prepare by thinking of the different code they will need, try their plan in Game Lab, then reflect on what they have created. They also have a chance to share their creations with their peers.
    - Sprite Scenes
    - Create a Background
    - Add Sprites
    - Add Text
    - Review Your Scene

MODULE 1

- **Lesson 11: The Draw Loop**
  - **DESCRIPTION:** In this lesson students are introduced to the draw loop, one of the core programming paradigms in Game Lab. To begin the lesson students look at some physical flipbooks to see that having many frames with different images creates the impression of motion. Students then watch a video explaining how the draw loop in Game Lab helps to create this same impression in their programs. Students combine the draw loop with random numbers to manipulate some simple animations with dots and then with sprites.
    - Video: Introduction to the Draw Loop
    - Exploration
    - Skill Building
    - Prediction
    - Skill Building
    - Practice
    - Assessment
    - Challenges

- **Lesson 12: Sprite Movement**
  - **DESCRIPTION:** In this lesson, students learn how to control sprite movement using a construct called the counter pattern, which incrementally changes a sprite's properties. Students first brainstorm different ways that they could animate sprites by controlling their properties, then explore the counter pattern in Code Studio. After examining working code, students try using the counter pattern to create various types of sprite movements.
    - Prediction
    - Video: Sprite Movement
    - Skill Building
    - Practice
    - Assessment
    - Challenges

- **Lesson 13: Mini-Project - Animation**
  - **DESCRIPTION:** In this lesson, students are asked to combine different methods that they have learned to create an animated scene. Students first review the types of movement and animation that they have learned, and brainstorm what types of scenes might need that movement. They then begin to plan out their own animated scenes, which they create in Game Lab.
    - Example Animated Scene
    - Plan Your Scene
    - Draw a Background
    - Add Sprites
    - Add Text
    - Add Movement
    - Review Your Animated Scene

- **Lesson 14: Conditionals**
  - **DESCRIPTION:** This lesson introduces booleans and conditionals, which allow a program to run differently depending on whether a condition is true. Students start by playing a short game in which they respond according to whether particular conditions are met. They then move to Code Studio, where they learn how the computer evaluates Boolean expressions, and how they can be used to structure a program.
    - Prediction
    - Video: Booleans
    - Quick Check
    - Skill Building
    - Video: Conditional Statements
    - Skill Building
    - Practice
    - Assessment
    - Challenges

- **Lesson 15: Keyboard Input**
  - **DESCRIPTION:** Following the introduction to booleans and if statements in the previous lesson, students are introduced to a new block called `keyDown()` which returns a boolean and can be used in conditionals statements to move sprites around the screen. By the end of this lesson, students will have written programs that take keyboard input from the user to control sprites on the screen.
    - Prediction
    - Skill Building
    - Practice
    - Assessment
    - Challenges

- **Lesson 16: Mouse Input**
  - **DESCRIPTION:** In this lesson students continue to explore ways to use conditional statements to take user input. In addition to the keyboard commands learned yesterday, students will learn about several ways to take mouse input. They will also expand their understanding of conditionals to include else, which allows for the computer to run a certain section of code when a condition is true, and a different section of code when it is not.
    - Prediction
    - Video: If/Else Statements
    - Skill Building
    - Practice
    - Assessment
    - Challenges

- **Lesson 17: Project - Interactive Card**
  - **DESCRIPTION:** In this culminating project for Chapter 1, students plan for and develop an interactive greeting card using all of the programming techniques they've learned to this point.
    - Demo App: Interactive Card
    - Make an Interactive Card
    - Project Work
    - Reflection

- **Lesson 18: Velocity**
  - **DESCRIPTION:** After a brief review of how they used the counter pattern to move sprites in previous lessons, students are introduced to the idea of hiding those patterns in a single block. Students then head to Code Studio to try out new blocks that set a sprite's velocity directly, and look at various ways that they are able to code more complex behaviors in their sprites.
    - Skill Building
    - Video: Velocity
    - Skill Building
    - Practice
    - Assessment
    - Challenges

- **Lesson 19: Collision Detection**
  - **DESCRIPTION:** Students learn about collision detection on the computer. Working in pairs, they explore how a computer could use sprite location and size properties and math to detect whether two sprites are touching. They then use the `isTouching()` block to create different effects when sprites collide, and practice using the block to model various interactions.
    - Sample Game
    - Skill Building
    - Practice
    - Assessment
    - Challenges

- **Lesson 20: Mini-Project - Side Scroller**
  - **DESCRIPTION:** Students use what they have learned about collision detection and setting velocity to create a simple side scroller game. After looking at a sample side scroller game, students brainstorm what sort of side scroller they would like to make, then use a structured process to program the game in Code Studio.
    - Intro to Side Scrollers
    - Draw Your Background
    - Create Your Sprites
    - Player Controls
    - Looping
    - Sprite Interactions
    - Scoring & Scoreboard
    - Review Your Game

- **Lesson 21: Complex Sprite Movement**
  - **DESCRIPTION:** Students learn to combine the velocity properties of sprites with the counter pattern to create more complex sprite movement. After reviewing the two concepts, they explore various scenarios in which velocity is used in the counter pattern, and observe the different types of movement that result. In particular, students learn how to simulate gravity. They then reflect on how they were able to get new behaviors by combining blocks and patterns that they already knew.
    - Prediction
    - Skill Building
    - Practice
    - Assessment
    - Challenges

- **Lesson 22: Collisions**
  - **DESCRIPTION:** In this lessson, students program their sprites to interact in new ways. After a brief review of how they used the `isTouching` block, students brainstorm other ways that two sprites could interact. They then use `isTouching` to make one sprite push another across the screen before practicing with the four collision blocks ((`collide`, `displace`, `bounce`, and `bounceOff`).
    - Code Prediction
    - Skill Building
    - Practice
    - Assessment
    - Challenges

Lesson 23: Mini-Project - Flyer Game

- **DESCRIPTION:** Students use what they have learned about simulating gravity and the different types of collisions to create simple flyer games. After looking at a sample flyer game, students brainstorm what sort of flyer games they would like, then use a structured process to program the game in Code Studio.
    - Intro to Flyer Game
    - Make Your Sprites
    - Player Controls
    - Sprite Movement
    - Sprite Interactions
    - Review Your Game

Lesson 24: Functions

- **DESCRIPTION:** Students learn how to create functions to organize their code, make it more readable, and remove repeated blocks of code. Students first think about what sorts of new blocks they would like in Game Lab, and what code those blocks would contain inside. Afterwards students learn to create functions in Game Lab. They will use functions to remove long blocks of code from their draw loop and to replace repeated pieces of code with a single function.
    - Video: Functions
    - Skill Building
    - Predict
    - Practice
    - Quick Check
    - Collector Game
    - Challenges

Lesson 25: The Game Design Process

- **DESCRIPTION:** This lesson introduces students to the process they will use to design games for the remainder of the unit. This process is centered around a project guide which asks students to define their sprites, variables, and functions before they begin programming their game. In this lesson students begin by playing a game on Game Lab where the code is hidden. They discuss what they think the sprites, variables, and functions would need to be to make the game. They are then given a completed project guide which shows one way to implement the game. Students are then walked through this process through a series of levels. At the end of the lesson, students have an opportunity to make improvements to the game to make it their own.
    - Same Game
    - Plan Your Project
    - Set Up Sprites
    - Control Your Player
    - Sprite Interactions
    - Challenges

Lesson 26: Using the Game Design Process

- **DESCRIPTION:** In this multi-day lesson, students use the problem solving process from Unit 1 to create a platform jumper game. They start by looking at an example of a platform jumper, then define what their games will look like. Next, they use a structured process to plan the backgrounds, variables, sprites, and functions they will need to implement their game. After writing the code for the game, students will reflect on how the game could be improved, and implement those changes.
    - Sample Platform Jumper Game
    - Build a Platform Jumper
    - Platform Jumper - Background and Variables
    - Platform Jumper - Platforms
    - Platform Jumper - Items
    - Platform Jumper - Player
    - Platform Jumper Review
    - Challenges

- **Lesson 27: Project - Design a Game**
  - **DESCRIPTION:** Students will plan and build their own game using the project guide from the previous two lessons. Working individually or in pairs, students will first decide on the type of game they'd like to build, taking as inspiration a set of sample games. They will then complete a blank project guide where they will describe the game's behavior and scope out the variables, sprites, and functions they'll need to build. In Code Studio, a series of levels prompts them on a general sequence they can use to implement this plan. Partway through the process, students will share their projects for peer review and will incorporate feedback as they finish their game. At the end of the lesson, students will share their completed games with their classmates. This project will span multiple classes and can easily take anywhere from 3-5 class periods.
    - Sample Games
    - Project - Background and Variables
    - Project - Sprites and Interactions
    - Review Your Game
    - End-Of-Unit Survey

- **Lesson 28: CS Discoveries Post-Course Survey**
    - Post-Course Survey Instructions

- **Post-Project Test**

UNIT 4

THE DESIGN PROCESS

DESCRIPTION

The Design Process unit transitions students from thinking about computer science as a tool to solve their own problems towards considering the broader social impacts of computing. Through a series of design challenges, students are asked to consider and understand the needs of others while developing a solution to a problem. The second half of the unit consists of an iterative team project, during which students have the opportunity to identify a need that they care about, prototype solutions both on paper and in App Lab, and test their solutions with real users to get feedback and drive further iteration.

Implementation Guidance for The Design Process

Complete the first three lessons of the Problem Solving and Computing unit before starting
You can view the full Curriculum Guide and Implementation Guide for more information about this unit

- **Lesson 1: Designing With Empathy**
  - **DESCRIPTION:** To kick off a unit devoted to group problem solving and developing products for other users, students begin by investigating the design of various shoes. Students look at a variety of shoes and attempt to match each shoe with a potential user. Students also provide feedback on these designs, using an “I Like…” / “I Wish…” / “I Wonder…” structure that will be repeated throughout the unit. To conclude the activity, students are asked to propose some changes to one set of shoes that would make it more useful or usable.

- **Lesson 2: Understanding Your User**
  - **DESCRIPTION:** Designers need to understand their users' needs in order to create useful products. This lesson encourages students to think about how to design for another person by role-playing as someone else using a user profile and reacting as that user to a series of products. Each student is assigned a user profile describing a person, which they then use to choose appropriate products, critique product design, and suggest improvements to design.

MODULE 1

- **Lesson 3: User Centered Design - Define and Prepare**
  - **DESCRIPTION:** This is the first part of a two-day lesson where students are guided through an abbreviated version of the design process they will be seeing throughout this unit. This lesson focuses on the Define and Prepare steps of the process. Students first brainstorm a list of potential users of smart clothing. As a class, they then group these ideas into broad categories and each group will choose one category of user. Groups repeat this process to brainstorm needs or concerns of their user, eventually categorizing these needs and choosing one to focus on. In the next lesson, students will complete the Try and Reflect steps of the design process.

- **Lesson 4: User Centered Design - Try and Reflect**
  - **DESCRIPTION:** This is the second part of a two-day lesson where students are guided through an abbreviated version of the design process they will be seeing throughout this unit. Yesterday students completed the Define and Prepare steps of the design process. Today, they complete the Try and Reflect portions. Students continue their work from yesterday by designing a piece of smart clothing, using the specific needs and concerns they brainstormed to guide their decision making. Students have a chance to share their decision-making process and get feedback on how well their product addresses the user needs they selected.

- **Lesson 5: User Interfaces**
  - **DESCRIPTION:** Following the mini design project, students look towards the next phase of design - prototyping a product that attempts to address user needs. In teams, students examine a paper prototype for a chat app called "Txt Ur Grndkdz". Through using this paper prototype, students get a chance to see how a simple paper prototype can be used to quickly test ideas and assumptions before we ever get to the computer. After "using" the provided prototype students begin to identify ways to improve the next iteration.

- **Lesson 6: Feedback and Testing**
  - **DESCRIPTION:** In this lesson, students use feedback from "users" of the paper-prototyped app from the previous lesson in order to develop improvements to the user interface of that paper prototype. The lesson begins with a reflection on the fact that designers need to translate human needs with technology into changes to the user interface or experience. Students are then given a collection of feedback and requests from users of the app from the previous lesson. In groups students categorize the feedback and identify ways the needs expressed in the feedback could be met by changes to the interface of the app. Then in groups students will implement some of these changes to meet one of the needs they identified.

- **Lesson 7: Identifying User Needs**
  - **DESCRIPTION:** In this lesson, students interpret user interviews to determine the needs & interests of a user. They then speculate on the barriers these users are facing or the opportunities that are available for each user. In a group, they share these barriers & opportunities and brainstorm different apps that could be used to address these issues. By the end of the lesson, students will have decided on an app idea that addresses a barrier or opportunity for a user.

- **Lesson 8: Project - Paper Prototype**
  - **DESCRIPTION:** Based on user interview from the previous lesson, each student comes up with an idea for an app that will address their user's problem. Students then get to create their own paper prototype of their app ideas by drawing "screens" on individual notecards. A project guide directs students through the process including building the paper prototype and testing it with their user to see if their assumptions about the user interfaces they created are accurate.

    This is a two-day lesson, with specific timings outlined in the lesson plan below

    - Reflection

- **Lesson 9: Designing Apps for Good**
  - **DESCRIPTION:** This lesson starts an app design project that lasts through the end of the unit. Students first explore a number of apps designed for social impact that have been created by other students. Students form project teams and lay out a contract for how the team will function throughout the development of their app. Finally, they brainstorm a team name, a topic for their app, and potential users.
    - Sample Apps for Good

- **Lesson 10: Market Research**
  - **DESCRIPTION:** In this lesson students research apps similar to the one they intend on creating to better understand the needs of their users. Students work within their teams to search the Internet for other apps, then evaluate the ones they find interesting. By the end of the lesson, each team will have a clearer idea about the type of app they want to create and further refine who their target users are. Each team will maintain a list of citations for all the apps they examined for use in their final presentation.

MODULE 1

- **Lesson 11: Exploring UI Elements**
  - **DESCRIPTION:** Before starting to design apps, we need to help students scope their expectations. Because students will eventually be prototyping these apps in App Lab, they will be in better shape if their ideas align with the kinds of apps that can be made using AppLab's tools. Today focuses on having teams look at several example apps made in AppLab and identifying the components of the user interface. They then make a plan for which features could use which components in their app.
    - Example Apps

- **Lesson 12: Build a Paper Prototype**
  - **DESCRIPTION:** Students work in their teams to create paper prototypes for the apps they've been developing. They begin by making a plan for each screen, then assigning different team members to work on each screen so the task can be completed in time. They can use 3x5 index cards to develop their paper prototypes, or the Paper Prototype template.

- **Lesson 13: Prototype Testing**
  - **DESCRIPTION:** The primary purpose of developing paper prototypes is that they allow for quick testing and iteration before any code is written. This lesson is focused on giving teams a chance to test their prototypes before moving to App Lab. Teams develop a plan to test with users before running prototype tests with multiple other students in the class (and potentially outside the class). In order to test the prototype with the users, the students will have to assign roles in the testing (the "narrator", the "computer" and the "observers") as well as have some questions prepared for the user to answer after the test is complete.

- **Lesson 14: Design Mode in App Lab**
  - **DESCRIPTION:** Students are introduced to the App Lab programming environment that they will use to build their apps. Students work through a progression of skill-building levels to learn how to use the drag-and-drop design mode of App Lab. They end the lesson by making a plan to adapt one of their Paper Prototype screens into App Lab, focusing on having unique IDs for each element.
    - Video: Intro to Design Mode
    - Build a Screen
    - Finish a Screen

- **Lesson 15: Build a Digital Prototype**
  - **DESCRIPTION:** Having collaboratively developed a paper prototype for their apps, groups now divide and conquer to begin work on an interactive digital version based on the paper prototype. Using the drag-and-drop design mode of App Lab, students individually work through a progression of skill-building levels to learn how to build digital versions of a paper prototype. From there, each group member builds out at least one page of their app in App Lab, to be later combined into a single app.
    - Design a Screen for your App
    - App Project: Screen Design

- **Lesson 16: Events in App Lab**
  - **DESCRIPTION:** In this lesson, students learn how to add screens to their apps, how to import screens created by other students, and how to program events to navigate between screens. Students learn basic event-driven programming by building up the model app that they started in the previous lesson. At the end of the lesson, students make a plan for how they will stay organized when importing each other's screens in tomorrow's lesson.
    - Video: Events in App Lab
    - Adding Screens
    - Events and Linking Screens

- **Lesson 17: Linking Prototype Screens**
  - **DESCRIPTION:** In this lesson, teams combine their digital prototype screens into a single app, which they can then link together using code. They share their project's import URLs with each other, then import each teammate's screen into their own app. Each student works independently to add events to their code to link the screens together, completing their digital prototype.
    - Combining Projects

- **Lesson 18: Testing the App**
  - **DESCRIPTION:** By the end of the previous lesson, each team should have a working prototype of their app. The primary purpose of this lesson is to have the team actually test the app with other people, preferably from the target audience the app is intended for, or from different teams in the class while observers from the team will record the results. As with testing the paper prototypes, teams will start by planning for the specific scenarios before running and observing tests.

- **Lesson 19: Bugs and Features**
    - **DESCRIPTION:** Teams have at this point developed an app prototype that has gone through multiple iterations and rounds of user testing. With the information and guidance gained from the last round of user testing, each student has the opportunity to plan for and implement improvements to the team app. Depending on the time you have available, and student interest, you can run the cycle of testing and iteration as many times as you see fit.

- **Lesson 20: Updating Your Prototype**
    - **DESCRIPTION:** Teams make a decision about how they will incorporate updates to their app, then they begin updating their app based on their post-its from yesterday's lesson. They use a "To Do / Doing / Done" chart to organize their work and track their progress. Teams spend the majority of the class working on updating their app, with time at the end to re-combine into a single app if necessary.
        - Updating Your Prototype

- **Lesson 21: Project - App Presentation**
    - **DESCRIPTION:** At this point, teams have researched a topic of personal and social importance, developed and tested both a paper prototype and a digital prototype, and iterated on the initial app to incorporate new features and bug fixes. Now is the time for them to review what they have done and pull together a coherent presentation to demonstrate their process of creation. Using the provided presentation template, teams prepare to present about their process of app development, including the problem they set out to solve, the ways in which they've incorporated feedback from testing, and their plans for the future.
        - End-Of-Unit Survey

- **Post-Project Test**

- **Lesson 22: CS Discoveries Post-Course Survey**
    - Post-Course Survey Instructions

UNIT 5

DATA AND SOCIETY

DESCRIPTION

The Data and Society unit is about the importance of using data to solve problems and it highlights how computers can help in this process. The first chapter explores different systems used to represent information in a computer and the challenges and tradeoffs posed by using them. In the second chapter, students learn how collections of data are used to solve problems, and how computers help to automate the steps of this process. In the final project, students gather their own data and use it to develop an automated solution to a problem.

Implementation Guidance for Data and Society

Complete the first three lessons of the Problem Solving and Computing unit before starting

You can view the full Curriculum Guide and Implementation Guide for more information about this unit

➢ **Lesson 1: Lesson 1: Representation Matters**

- **DESCRIPTION:** In the first lesson of the data unit, students get an overview of what data is and how it is used to solve problems. Students start off with a brief discussion to come to a common understanding of data. They then split into groups and use a data set to make a series of meal recommendations for people with various criteria. Each group has the choices of meal represented in a different way (pictures, recipes, menu, nutrition) that gives an advantage for one of the recommendations. Afterwards, groups compare their responses and discuss how the different representations of the meal data affected how the students were able to solve the different problems.

➢ **Lesson 2: Patterns and Representation**

- **DESCRIPTION:** In this lesson students create their own system for representing information. They begin by brainstorming all the different systems they already use to represent yes-no responses. They then work in small groups to create a system that can represent any letter in the alphabet using only a single stack of cards. The cards used have one of 6 different possible drawings (6 animals, 6 colors, etc.) and so to represent the entire alphabet students will need to use patterns of multiple cards to represent each letter. Students create messages with their systems and exchange with other groups to ensure the system worked as intended. In the wrap-up discussion the class reviews any pros and cons of the different systems. They discuss commonalities between working systems and recognize that there are many possible solutions to this problem and what's important is that everyone uses the same arbitrary system to communicate.

- **Lesson 3: ASCII and Binary Representation**
  - **DESCRIPTION:** In this lesson students learn to use their first binary system for encoding information, the ASCII system for representing letters and other characters. At the beginning of the lesson the teacher introduces the fact that computers must represent information using either "on" or "off". Then students are introduced to the ASCII system for representing text using binary symbols. Students practice using this system before encoding their own message using ASCII. At the end of the lesson a debrief conversation helps synthesize the key learning objectives of the activity.

- **Lesson 4: Representing Images**
  - **DESCRIPTION:** In this lesson students learn how computers represent images. To begin the lesson they consider the challenge of turning all the complexity of vision into a binary pattern. Through a series of images showing how this transformation is made students are introduced to the concept of splitting images into squares or "pixels" which can then be turned on or off individually to make the entire image. Students then do a short set of challenges using the Pixelation Widget in order to draw black and white images. Puzzles are designed to call out some of the challenges of representing images in this way. In the wrap up students make connections between the system for representing images and the system for representing text they learned in the previous lesson.
    - Representing Images

- **Lesson 5: Representing Numbers**
  - **DESCRIPTION:** In this lesson, students learn about the binary number system. With a set of cards that represent the place values in a binary (base-2) number system by a collection of dots, students turn bits "on" or "off" by turning cards face up and face down, then observe the numbers that result from these different patterns. Eventually, students extend the pattern to a generic 4-bit system.
    - Binary Number Widget
    - Binary and Data

- **Lesson 6: Combining Representations**
  - **DESCRIPTION:** In this lesson, students use all three types of binary representation systems (ASCII characters, binary number, and images) to decode information in a record. After seeing a series of bits and being asked to decode them, students are introduced to the idea that in order to understand binary information, they must understand both the system that is being used and the meaning of the information encoded. They then decode a record representing a pet based on a given structure.
    - 8-Bit Binary Number Widget

- **Lesson 7: Keeping Data Secret**
    - **DESCRIPTION:** Students continue to explore how data is represented in a punchcard, and begin considering whether some data should be protected from public view because it is too personal or sensitive. Once students understand the reasons for protecting data, they learn a binary encryption system that lets them encrypt and decrypt data in their punchcards, and eventually they are able to send secret messages to one another using this method. Class concludes with a discussion on the importance of protecting our information and how encryption is one way we can accomplish this.

- **Lesson 8: Project - Create a Representation**
    - **DESCRIPTION:** In this lesson students design a structure to represent their perfect day using the binary representation systems they've learned in this chapter. Students will first write a short description of their perfect day and then review with a partner to identify the key pieces of information they think a computer could capture. As a class students will decide how a punch card of bytes of information will be interpreted to represent those pieces of information. Students will then use the ASCII, binary number, and image formats they have learned to represent their perfect days. Students then trade punch cards and try to decode what the other student's perfect day is like. The lesson ends with a reflection.
        - Reflection

- **Lesson 9: Problem Solving With Data**
    - **DESCRIPTION:** In this lesson, students use the problem solving process from earlier in the course to solve a data problem. After reviewing the process, the class is presented with a decision: whether a city should build a library, pet shelter, or fire department. Students work in teams to collect information on the Internet to help them decide what should be built, then use this information to build an argument that will convince the city council of their choice. They then map what they have done to the problem solving process that they have been using throughout the course, comparing the general problem solving process to its specific application to data problems.

- **Lesson 10: Structuring Data**
    - **DESCRIPTION:** In this lesson, students go further into the collection and interpretation of data, including cleaning and visualizing data. Students first look at how presenting data in different ways can help people to understand it better, and they then create visualizations of their own data. Using the results of a preferred pizza topping survey, students must decide what to do with data that does not easily fit into the visualization scheme that they have chosen. Finally, students discuss which parts of this process can be automated by a computer and which need a human to make decisions.
        - Pizza Party Data

MODULE 1

- **Lesson 11: Interpreting Data**
    - **DESCRIPTION:** Students begin the lesson by looking at a cake preference survey that allows respondents to specify both a cake and an icing flavor. They discuss how knowing the relationship between cake and icing preference helps them better decide which combination to recommend. They are then introduced to cross tabulation, which allows them to graph relationships to different preferences. They use this technique to find relationships in a preference survey, then brainstorm the different types of problems that this process could help solve.

- **Lesson 12: Making Decisions with Data**
    - **DESCRIPTION:** In this lesson students get practice making decisions with data based on problems designed to be familiar to middle school students. Students work in groups discussing how they would use the data presented to make a decision before the class discusses their final choices. Not all questions have right answers and in some cases students can and should decide that they should collect more data. The lesson concludes with a discussion of how different people could draw different conclusions from the same data, or how collecting different data might have affected the decisions they made.

- **Lesson 13: Automating Data Decisions**
    - **DESCRIPTION:** In this lesson students look at a simple example of how a computer could be used to complete the decision making step of the data problem solving process. Students are given the task of creating an algorithm that could suggest a vacation spot. Students then create rules that a computer could use to make this decision automatically. Students share their rules and what choices their rules would make with the class data. They then use their rules on data from their classmates to test whether their rules would make the same decision that a person would. The lesson concludes with a discussion about the benefits and drawbacks of using computers to automate the data problem solving process.

- **Lesson 14: Problem Solving with Big Data**
    - **DESCRIPTION:** In this lesson, students look at how data is collected and used by organizations to solve problems in the real world. Students are presented with three scenarios that could be solved using data and brainstorm the types of data they would need and how they could collect the data. Each problem is designed to reflect a real-world service that exists. After brainstorming, students watch a video about a real-world service and record notes about what data is collected by the real-world service and how it is used. At the end of the lesson, students record whether data was provided actively by a user, was recorded passively, or is collected by sensors.
        - Video: Netflix and Data
        - Video: Waze and Data
        - Video: Amazon and Data

MODULE 1

- **Lesson 15: Data and Machine Learning**
  - **DESCRIPTION:** In this lesson, students are introduced to the concepts of Artificial Intelligence and Machine Learning using the AI for Oceans widget. First students classify objects as either "fish" or "not fish" to attempt to remove trash from the ocean. Then, students will need to expand their training data set to include other sea creatures that belong in the water. In the second part of the activity, students will choose their own labels to apply to images of randomly generated fish. This training data is used for a machine learning model that should then be able to label new images on its own.
    - Video: What is Machine Learning?
    - AI for Oceans: Fish vs Not Fish
    - Video: Training Data and Bias
    - AI for Oceans: Types of Fish
    - Video: Impact on Society

- **Post-Project Test**

- **Lesson 16: Project - Make a Recommendation**
  - **DESCRIPTION:** To conclude this unit, students design a recommendation engine based on data that they collect and analyze from their classmates. After looking at an example of a recommendation app, students follow a project guide to complete this multi-day activity. In the first several steps, students choose what choice they want to help the user to make, what data they need to give the recommendation, create a survey, and collect information about their classmates' choices. They then interpret the data and use what they have learned to create the recommendation algorithm. Last, they use their algorithms to make recommendations to a few classmates. Students perform a peer review and make any necessary updates to their projects before preparing a presentation to the class.
    - Sample App
    - End-Of-Unit Survey

- **Lesson 17: CS Discoveries Post-Course Survey**
    - Post-Course Survey Instructions

UNIT 6

PHYSICAL COMPUTING

DESCRIPTION

In the Physical Computing unit, students explore the role of physical devices in computing. Using App Lab and Adafruit's Circuit Playground, students develop programs that utilize the same hardware inputs and outputs that you see in the smart devices, looking at how a simple rough prototype can lead to a finished product. Then, students explore how physical devices can be used to react to the world around them using a "maker" mindset to create prototypes with everyday materials.

Implementation Guidance for Physical Computing

Complete the first three lessons of the Problem Solving and Computing unit before starting

This unit assumes students are familiar with App Lab, specifically using Design Mode to create a user interface and using Events to change the screen when clicking a button. This can be accomplished by having students complete the App Lab Hour of Code before starting this unit.

This unit requires students to have access to Circuit Playground's - click here for more information about using Circuit Playground's in your classroom

Guidance for how to support students in programming levels and differentiate tasks are available in the Programming Levels Guide and Differentiation Guide

You can view the full Curriculum Guide and Implementation Guide for more information about this unit

- **Lesson 1: Innovations in Computing**
  - **DESCRIPTION:** In this lesson, students will do some research into interesting innovations in computing. This lesson will expose students to a wider variety of computing form factors (what a computer looks like) and fields that are impacted by computing. Later in this unit students will look back on the devices they encountered in this lesson as they develop their own physical computing devices.
  - Question of the Day: What is a computing innovation?

- **Lesson 2: The Circuit Playground**
  - **DESCRIPTION:**In this lesson students get their first opportunity to write programs that use the Circuit Playground. After first inspecting the board visually and hypothesizing possible functionalities, students move online where they will learn to write applications that control an LED and use a buzzer to make sounds. By combining App Lab screens with the Circuit Playgrounds, students can gradually start to integrate elements of the board as an output device while relying on App Lab for user input.
  - Question of the Day: What outputs can I control on the circuit playground?
    - Setup
    - Predict
    - Skill Building
    - Predict
    - Skill Building
    - Practice
    - Assessment
    - Challenge

Lesson 3: Updating Screen Elements

- **DESCRIPTION:** Students should already be familiar with programming in App Lab, specifically using Design Mode to create screen elements and using `onEvent` blocks to create button click events. To expand the kinds of apps that students can make, and to encourage them to think in new ways about how users interact with apps, we introduce the `setProperty()` and `setText()` blocks that allow users to change the properties and content of various UI elements. In this lesson students explore how they can use events to update elements on the screen

- Question of the Day: How can you update design elements of an app?
  - Predict
  - Skill Building
  - Practice
  - Assessment
  - Challenge

Lesson 4: Board Events

- **DESCRIPTION:** This lesson transitions students from considering the Circuit Playground as strictly an output device and instead introduces the buttons and toggle switches as tools for input. Starting with the hardware buttons and switch, students learn to use `onBoardEvent()`, analogously to `onEvent()`, in order to take input from their Circuit Playgrounds.

- Question of the Day: How can the user interact with the circuit playground for input?
  - Predict
  - Skill Building
  - Practice
  - Assessment
  - Challenge

Lesson 5: Variables and If Statements

- **DESCRIPTION:** In this lesson, students are introduced to variables, the counter pattern, and if-statements. Students will use these concepts in the context of programming the circuit playground and creating more complex input/output behaviors, such as counting the number of button presses before having the circuit playground make a noise. The concepts in this lesson are used to make more complex and fulfilling apps in future lessons, and students may need to refer back to the videos and examples in this lesson as they continue to master these concepts.

- Question of the Day: How can I create more complex programs that keep track of information and make decisions?
  - Predict
  - Skill Building
  - Video - Variables
  - Skill Building
  - Video - The Counter Pattern
  - Skill Building
  - Video - If Statements
  - Predict
  - Skill Building
  - Practice
  - Assessment
  - Challenge

Lesson 6: Mini-Project: Field Collector App

- **DESCRIPTION:** In this mini-project, students will use the Circuit Playground to collect data, then use App Lab to analyze the data that was collected. This is similar to citizen science fieldwork or survey apps where a person collects data on a device "in the field", then returns to a central location to view and analyze the data (such as cataloguing endangered species in the wild). Students will use variables and events to collect data from the circuit playground, then use if-statements to make decisions or recommendations based on the data they collect.

- Question of the Day: How can I use events and variables to create an app that collects and analyzes data?
  - Field Collector App

- **Lesson 7: Color LEDs**
  - **DESCRIPTION:** In this lesson, students learn how to use the 10 color LEDs on the Circuit Playground. Students will control the color and intensity of each LED, then use what they have learned to program light patterns to create a light show on their Circuit Playground.
  - Question of the Day: How can you use the colored LEDs in an app?
    - Predict
    - Skill Building
    - Practice
    - Assessment
    - Challenge

- **Lesson 8: Getting Screen Inputs**
  - **DESCRIPTION:** Students learn to use several new design elements - text inputs, dropdowns, and sliders - so they can get user input from the screen of their apps. This lesson also introduces students to the getProperty and getText blocks, which allow them to access their user input in their code. Students first practice using these tools to determine what the user has input in various user interface elements. Students later use getProperty and setProperty together with the counter pattern to make elements move across the screen. A new event trigger, change, is also introduced to represent when a dropdown or slider changes values.
  - Question of the Day: How can we use the app screen for user input?
    - Predict
    - Skill Building
    - Practice
    - Assessment
    - Challenge

- **Lesson 9: Combining Inputs and Outputs**
  - **DESCRIPTION:** In this lesson, students combine inputs and outputs across both the circuit playground and the app screen. For example, students may use dropdowns or sliders to make the LEDs on the circuit playground change. Or, students may use the buttons on the circuit playground to update elements on the screen. This lesson also extends how students use variables and if-statements to make more complex decisions based on user input. There are no new coding commands or circuit playground features in this lesson, but students are combining many elements from previous lessons in new ways.
  - Question of the Day: How can I use inputs and outputs from both the circuit playground and the app screen?
    - Predict
    - Skill Building
    - Practice
    - Assessment
    - Challenge

- **Lesson 10: Project: Human Device Interaction**
  - **DESCRIPTION:** In this project, students create an app that controls the Circuit Playground so it interacts with the physical environment around it, similar to many smart phone apps that are used to control devices in a house or car or school. Students use physical materials to help design their physical device, then create an app that lets the user interact with the physical device or change settings.
    - Human Device Interaction
    - Reflection

MODULE 1

- **Lesson 11: Board Sensors**
  - **DESCRIPTION:** In this lesson, students explore how the three sensors (sound, light, and temperature) can be used to write programs that respond to changes in the environment. The use of these sensors marks a transition in terms of how users interact with a program. By using sensors as an input, the user of an app doesn't have to directly interact with it at all or may interact without actually realizing they are doing so.
  - Question of the Day: How can I use sensors to detect changes in the environment?
    - Predict
    - Skill Building
    - Predict
    - Skill Building
    - Practice
    - Assessment
    - Challenge

- **Lesson 12: Accelerometer**
  - **DESCRIPTION:** In this lesson, students will explore the accelerometer and its capabilities. They'll become familiar with its events and properties, as well as create multiple programs utilizing the accelerometer similar to those they've likely come across in real world applications.
  - Question of the Day: How can I use the accelerometer to create apps?
    - Predict
    - Skill Building
    - Practice
    - Assessment
    - Challenge

- **Lesson 13: Making Music**
  - **DESCRIPTION:** In this lesson students will use the buzzer to its full extent by producing sounds, notes, and songs with the buzzer. Students start with a short review of the buzzer's frequency and duration parameters, then move on to the concept of notes. Notes allow students to constrain themselves to frequencies that are used in Western music and provide a layer of abstraction that helps them to understand which frequencies might sound good together. Once students are able to play notes on the buzzer, they use arrays to hold and play sequences of notes, forming simple songs.
  - Question of the Day: How can I use the buzzer to play musical notes or songs?
    - Predict
    - Investigate
    - Skill Building
    - Predict
    - Video - Introduction to Arrays
    - Skill Building
    - Assessment
    - Challenge

- **Lesson 14: Functions**
  - **DESCRIPTION:** This lesson introduces students to functions as way to organize and group repeated blocks of code together, such as changing all of the LEDs to red or blue. Then, students learn how to use parameters in their functions as a way to generalize behaviors to work for different contexts - for example, changing all the LEDs to a certain color rather than always changing them red or blue. Students will use these concepts in future lessons, and it may be necessary to return to this lesson to reference the resources and videos.
  - Question of the Day: How can I use functions to organize and simplify my code?
    - Predict
    - Video - Functions
    - Skill Building
    - Predict
    - Video - Parameters
    - Skill Building
    - Practice
    - Assessment

- **Lesson 15: Mini-Project: Interactive Art**
  - **DESCRIPTION:** In this lesson, students create a piece of interactive artwork using the sensors on the Circuit Playground and physical materials. Students explore how the Circuit Playground can augment physical materials to create an interactive experience. This project does not use the App Lab screen except to help with debugging - otherwise, users only interact with the physical artwork itself.
    - Interactive Art

- **Lesson 16: Physical Outputs and LEDs**
  - **DESCRIPTION:** In this lesson, students learn how to attach external LEDs to their circuit playground and use code to light up these LEDs. This allows students to create more flexible devices that mimic real-world products. This lesson requires several external materials, as well as safety protocols to ensure students don't accidentally damage their circuit playground.
  - Question of the Day: How can I use external LEDs to create physical prototypes?
    - Setup
    - Predict
    - Skill Building
    - Practice
    - Assessment
    - Challenge

- **Lesson 17: Physical Inputs and Buttons**
  - **DESCRIPTION:** In this lesson, students learn how to connect external wires to create input events when the wires touch, simulating a button press. Students learn to use code to recognize these external button events to make changes to their app. This allows students to create more flexible devices that mimic real-world products. This lesson requires several external materials, as well as safety protocols to ensure students don't accidentally damage their circuit playground.
  - Question of the Day: How can I use external buttons to create physical prototypes?
    - Setup
    - Skill Building
    - Practice
    - Assessment
    - Challenge

- **Lesson 18: Project - Prototype an Innovation**
  - **DESCRIPTION:** In this final project for the course, students team to develop and test a prototype for an innovative computing device based on the Circuit Playground. Using the inputs and outputs available on the board, groups will create programs that allow for interesting and unique user interactions.
    - Prototype an Innovation
    - End-Of-Unit Survey

MODULE 1

➢ **Post-Project Test**

➢ **Lesson 19: CS Discoveries Post-Course Survey**

- Post-Course Survey Instructions

UNIT 7

AI AND MACHINE LEARNING

DESCRIPTION

This unit is a hands-on introduction to developing a machine learning model with tabular data. Students explore how computers learn from data to make decisions, then develop machine learning projects around real-world data. The unit culminates in designing a machine learning app to solve a personally relevant problem.

Implementation guidance for the AI and Machine Learning Unit

Complete the first three lessons of the Problem Solving and Computing unit before starting

This unit assumes students are familiar with App Lab, specifically using Design Mode to create a user interface and using Events to change the screen when clicking a button. This can be accomplished by having students complete the App Lab Hour of Code before starting this unit.

Guidance for how to support students in programming levels and differentiate tasks are available in the Programming Levels Guide and Differentiation Guide

You can view the full Curriculum Guide and Implementation Guide for more information about this unit

Professional Development: If you are interested in teaching this course, we recommend completing the free AI/ML Self-Paced Professional Development Module. The module is designed to take 2 hours to complete. No previous experience with coding or AI is assumed.

- ➢ **Lesson 1: Introduction to Machine Learning**
  - ○ **DESCRIPTION:** In this lesson students are introduced to a form of artificial intelligence called machine learning and how they can use the Problem Solving Process to help train a robot to solve problems. They participate in three machine learning activities where a robot - A.I. Bot - is learning how to detect patterns in fish.
  - ○ Question of the Day: How can we use the Problem Solving Process to solve a problem with machine learning?
    - ■ Recognizing Fish
    - ■ Recognizing Fish Features
    - ■ Recognizing Expressions
    - ■ Check-In Survey

- ➢ **Lesson 2: Types of Machine Learning**
  - ○ **DESCRIPTION:** In this lesson students will consider how they create “mental” models when learning new concepts, and how those can be similar to a “machine learning” model. They participate in a color pattern activity to simulate building a machine learning model without help, then they play a game called "Green Glass Door" as an example of supervised learning, and finally, they will sort several scenarios into “supervised” or “unsupervised” learning.
  - ○ Question of the Day: What are different types of machine learning?
    - ■ Looking for Patterns
    - ■ Green Glass Door

MODULE 1

Lesson 3: Innovations in AI

- **DESCRIPTION:** In this lesson, students explore an application of AI called Seeing AI and examine how it is supporting people with visual impairments. Then, students research other examples of how AI is impacting society, focusing on users who are impacted by the examples they find. Finally, students share their findings with each other.
- Question of the Day: How is AI helping to solve problems around the world?

Lesson 4: Patterns in Data

- **DESCRIPTION:** In this lesson students will examine several apps that make decisions about what shoes to wear, ultimately building up to an understanding of how machine learning can help make this decision. Students are guided to the conclusion that surveying their users can help them make the best decision by looking for patterns in the data and basing their decisions on these patterns.
- Question of the Day: What strategies do computer models use to make decisions?
  - Shoe Recommender
  - Improved Shoe Recommender
  - Shoe Recommendation Survey
  - Ultimate Shoe Recommender

Lesson 5: Classification Models

- **DESCRIPTION:** In this lesson students will participate in an unplugged activity simulating one of the machine learning algorithms computers use to separate data into groups to help make decisions. Students will be tasked with helping a computer learn to classify food as fruits or vegetables, graph 20 different fruits on two axes comparing “sweetness” to “easy to eat”, and then try to separate the data into groups - a fruit area, and a veggie area.
- Question of the Day: How do computers learn to classify data?

Lesson 6: Introduction to AI Lab

- **DESCRIPTION:** In this lesson students will dive into the AI Lab tool for the first time, where they select features to train a model that predicts a given label. They start by exploring AI Lab and training a model to recognize shapes. Then they pretend they have been hired by several restaurants who would like to make recommendations to new customers based on survey data they’re collected, go through each dataset, and use data visualization tools to identify features with high relationships in the data.
- Question of the Day: How can we use machine learning to make recommendations?
  - Fruit and Veggie Explorer
  - Recognizing Shapes
  - Pizza Recommendations
  - You Choose: Restaurant Recommendations

MODULE 1

Lesson 7: Importing Models in App Lab

- **DESCRIPTION:** In this lesson students are introduced to importing their models into App Lab and linking their model to their screens. They help create a book recommendation app and learn how to add a welcome screen and events to their code. This lesson assumes students are already familiar with App Lab - for classrooms that have not seen App Lab before, consider extending this lesson and including additional videos or activities that are recommended in the lesson plan.
- Question of the Day: How can I create an app using machine learning?
  - Book Recommender App
  - Import a Model
  - Customize Your App
  - Choice Levels: Customize Your App

Lesson 8: Model Cards

- **DESCRIPTION:** In this lesson, students will investigate a model for bias and be introduced to a Model Card, which is a way of representing important information about a trained model that could help uncover bias. They will be investigating a Medical Priority app, which helps a hospital decide how soon to view patients based on their symptoms. As students go through the activity, they realize that the app is biased based on personal information and examine how this could happen.
- Question of the Day: How can we evaluate machine learning models once they've been trained?
  - Medical Priority App
  - Book Recommender Model Card
  - Medical Dataset Model Cards

Lesson 9: Saving Models in AI Lab

- **DESCRIPTION:** Students complete the full process of training and saving a model, then importing into App Lab. For the first time, students are able to choose the label they would like to predict and spend time deciding the features they will use to help predict their label of choice. Students also create a model card for their models in order to save them and import it into App Lab
- Question of the Day: How can I use Model Cards to document my decisions when training a machine learning model?
  - Zoo Predictions
  - Importing Your Model
  - Customizing Your App

Lesson 10: Model Cards in App Lab

- **DESCRIPTION:** In this lesson, students practice importing their models into App Lab, this time including models that have numerical data and using model cards to help improve the user experience of filling out their form. They will then learn how to view the model card within App Lab and use this to add more descriptive elements to an app. Next, they focus on improving the user experience by adding informational text to help guide users through completing the form and adding a style to their app to improve the user experience.
- Question of the Day: How can I use a Model Card to improve my app?
  - Example App: Zoo Classification
  - Pet Recommender
  - Adding New Screens
  - Practice
  - Assessment
  - Challenge

➢ Lesson 11: Numerical Models

- **DESCRIPTION:** In this lesson, students participate in an unplugged activity simulating a zombie outbreak. Students must predict which parts of town have the least amount of zombies using data from a neighboring town. Students will use degrees of similarity and averages to make predictions about the number of zombies at a particular location. Then, students are rescued and get to compare their predictions to the actual numbers as a way to discuss how accuracy is different for numerical data compared to categorical data.
- Question of the Day: How do computers learn to make predictions with numerical data?

➢ Lesson 12: Numerical Data in AI Lab

- **DESCRIPTION:** In this lesson, students will be introduced to numerical data which represents a range of values. Students are presented with a scenario where every feature and label is represented with numerical data, and they learn to use the new data visualization tools within AI Lab to help find patterns.
- Question of the Day: How can we use AI Lab to predict numerical data?
  - Safari Model
  - Safari App

➢ Lesson 13: Customizing Apps

- **DESCRIPTION:** In this lesson, students will explore how to customize the code of their app to make additional changes to the design of their app. They will start by exploring a single-screen app and then practice expanding the app to two-screens and updating the code to use the new design mode elements. After this, students help create a Driver Alert app that requires changes to the code using new design mode elements. Using the skills from this lesson, students will be able to create multi-screen apps where questions can appear on multiple screens instead of a single screen.
- Question of the Day: How can I customize the code for a machine learning app?
  - Raspado Recommender
  - Driver Alert System
  - You Choose: Improve Your App

➢ Lesson 14: AI Code of Ethics

- **DESCRIPTION:** In this one or two day project, students apply their skills from the unit so far and create a machine learning app using real-world data. Students are provided with several real-world datasets from a variety of contexts, and they choose which dataset they would like to investigate. They train and save their model, then make a simple App Lab app that uses the model. This mini-project is an opportunity to assess how well students can use features to create accurate machine learning models, and how well they can create apps that use machine learning.
- Question of the Day: Can I use real-world data to create an app that uses machine learning?
  - Train a Model in AI Lab
  - Create an App in App Lab
  - Reflection

- **Lesson 16: Issue Statements**
  - **DESCRIPTION:** This is the first of a five-day sequence of lessons that prepare students for the final project. In this lesson, students meet a team of fictional students who want to use machine learning to address an issue in their community. Students participate in an issue brainstorm using the 5 Why's strategy, then they help evaluate the ideas that the other student team came up with. The steps students take in this lesson are identical to the steps students will take in their final project.
  - Question of the Day: How can machine learning be used to address an issue in your community?

- **Lesson 22: CS Discoveries Post-Course Survey**
  - **DESCRIPTION:** This is the third in a five-day sequence of lessons that prepare students for the final project. In this lesson, students learn how to view survey data in Google Sheets and save the data to their computer as a csv file. Then, they upload the saved data to AI Lab and examine the survey results from one of the students to train a model using their data. Then, students use Google Sheets to examine data from another student where the data has errors and then try to fix the errors. The steps students take in this lesson are identical to the steps students will take in their final project, and the problem-solving strategies they develop will help them overcome challenges in their own final project.
  - Question of the Day: How can I import data into AI Lab to train a machine learning model?
    - Kim's Survey
    - Kim's Survey Data
    - AI Lab and Kim's Data
    - Isaac's Survey Data
    - AI Lab and Isaac's Data

- **Lesson 17: Survey Planning**
  - **DESCRIPTION:** This is the second in a five-day sequence of lessons that prepare students for the final project. In this lesson, students learn that the other team of students would like to create a club recommender app based on the clubs at their school. Students imagine what questions would be most useful to help make this recommendation, then they learn how to use a Google Form template to create a survey. The steps students take in this lesson are identical to the steps students will take in their final project.
  - Question of the Day: How can I create a survey to gather data for a machine learning app?
    - Create a Google Form

- **Lesson 19: Troubleshooting Models**
  - **DESCRIPTION:** This is the fourth of a five-day sequence of lessons that prepare students for the final project. In this lesson, students examine survey data from other members of the student team and analyze why their models are not working correctly. In examining the data, students develop strategies for avoiding these issues in the future and strategies for coping with these issues should they happen again. These are skills students will use in the final project as they develop their own surveys and collect data.
  - Question of the Day: What are strategies to make sure our data generates an accurate model?
    - Club Datasets
    - Alternative Datasets

- **Lesson 20: Creating an App**
  - **DESCRIPTION:** This is the fifth of a five-day sequence of lessons that prepare students for the final project. In this lesson, students import the club recommender app into App Lab and begin customizing the app. Students add a welcome screen and update the descriptions of each feature, then they can decide how they would like to further customize the app. The steps students take in this lesson are identical to the steps students will take in their final project.
  - Question of the Day: How can I create a friendly, easy-to-use machine learning app?
    - Import Your Model
    - Add a Welcome Screen
    - Customize Your App
    - Review Your App

- **Lesson 22: CS Discoveries Post-Course Survey**
    - Post-Course Survey Instructions

- **Lesson 21: Project - Design an AI App**
  - **DESCRIPTION:** To conclude this unit, students develop an AI app that addresses the social issue of their interest. Students follow a project guide to complete this multi-day activity. In the first step, students prepare the data they will use to train their model in AI Lab. After training, testing, and generating a model card, they export their model into App Lab for development. Here they use their model to create a user-friendly app. Students perform a peer review and make any necessary updates to their projects while reflecting on the outcome.
  - Question of the Day: How can I create an AI App the solves a problem in my community?
    - Design an AI App - AI Lab
    - Design an AI App - App Lab
    - End-Of-Unit Survey

MODULE 1

UNIT 8

CS DISCOVERIES POST-COURSE SURVEY

➢ **Post-Course Survey**

  ○ Survey