

Lesson 8: Project - Propose an App

225 minutes

Overview

To conclude this unit, this project combines the two major themes of Unit 1, the problem-solving process and the input/output/store/process model of a computer, to have students identify real-world problems and find ways to use technology to help solve them. This project will be completed across multiple days and will result in students creating a poster of a proposed app they design to solve a real-world problem, 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.

Question of the Day: How can the IOSP model help us to design an app that solves a problem?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming
- ▶ **CS** - Computing Systems

Agenda

Warm Up (5 minutes)

Introduce the Project

Activity (215 minutes)

Project Guide

Presenting Apps

Wrap Up (5 minutes)

Reflection

Teacher End-Of-Unit Survey

Extended Learning

Shark Tank

Post-Project Test

Objectives

Students will be able to:

- Design an app that inputs, outputs, stores, and processes information in order to solve a problem
- Identify and define a problem that could be solved using computing
- Provide and incorporate targeted peer feedback to improve a computing artifact

Preparation

- Print a copy of [Apps and Problem Solving](#) for each pair of students
- Poster paper, pens, markers and other supplies for making posters
- Check the ["Teacher's Lounge"](#) forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the teachers

- [Project - Propose an App](#) - Slides
▼ Make a Copy

For the students

- [Apps and Problem Solving](#) - Activity Guide
▼ Make a Copy
- [Apps and Problem Solving](#) - Peer Review
▼ Make a Copy

- [Apps and Problem Solving](#) - Rubric
▼ Make a Copy
- [Apps and Problem Solving - Student Checklist](#) - Resource
▼ Make a Copy
- [Computer Science Practices](#) - Activity Guide
▼ Make a Copy

Teaching Guide

Warm Up (5 minutes)

Introduce the Project

Prompt: Of the apps we've seen in this unit, what was your favorite? What problem did it solve?

Share: Allows students to share out their favorite apps.

Remarks

This whole unit we've been learning about the problem solving process and how computers process information to help solve problems by taking inputs and processing them to create useful outputs. Today we're going to start a project where you and a partner will design an app to solve a problem of your choosing.

Question of the Day: How can the IOSP model help us to design an app that solves a problem?

Activity (215 minutes)

Project Guide

Distribute: Give one project guide and one rubric or student checklist to each student. As a class, review the information provided on the first sheet of the project guide which explains the project, lists the steps, and shows what students will need to produce. Then provide a brief overview of each of the more detailed steps.

Step 1: Choose a Partner: Place students in pairs or groups of three.

Teaching Tip

Creating Groups: Ideally, this project is done in pairs. If need be, groups of 3 will work. You should decide beforehand whether you will assign or allow students to pick their partners.

Facilitating Group Projects: If students are working in pairs or small teams to complete projects, consider showing these two videos to the class:

- [How Teamwork Works](#)
- [Dealing with Disagreements](#)

Depending on your goals with this project, consider having teams complete a [Student Guide to Team Planning](#), which reinforces the message in the video

Step 2: Brainstorm Problems: Prompt groups to spend several minutes silently brainstorming problems and recording them on their project guides. Circulate the room and remind them that at this point they shouldn't be thinking about an app they want to build or even whether it's possible to solve this problem with an app. Make sure they're beginning with the problem rather than the solution.

💡 Teaching Tip ▲

Helping the Brainstorm: Listen carefully to student conversations. Identify students who are stuck, and the reason why they are stuck. Are they thinking too big? Help those students to think about problems as annoyances or inconveniences, or an opportunity to improve your quality of life to a small or big degree.

Step 3: Choose Your Problem: The project guide provides several criteria students can use to assess which of their problems they'd like to address. Ask students to look forward to Step 4 if they need more guidance on how they'll need to define or scope their problems. Give students a few minutes to discuss with their group and choose the problem they'd like to address with their app.

Step 4: Define Your Problem: For this step students will need to appropriately scope their problem by defining who their audience is, what specifically is the problem, and how they will know they have fixed it.

💡 Teaching Tip ▲

How Much to Help: At this point students have had a lot of practice defining problems. They also have a peer review process shortly after this step. Encourage them to be as detailed as possible but avoid giving specific advice for how to define their problem.

Step 5: Your App: Once students have scoped their problem, ask them to discuss an app that could be used to help solve their problem. To begin they'll just need to provide a high level description of the app that describes how a user would use it and what it does.

💡 Teaching Tip ▲

What Kind of App?: This project is supposed to result in a simple app, along the lines of those seen in the previous lesson. Even large problems or parts of large problems can be addressed by collecting and processing information appropriately.

Scoping Student Projects: Students may ideate projects that are beyond the skills they currently have or that would take longer than the allotted time to implement. Rather than asking students to choose a different project, consider asking students to imagine a more scaled-down version of their initial idea. As an analogy, if students initial idea is the "Run" step, imagine a less intense version that represents what the "Walk" step would look like. If necessary, you can keep going back further to a "Crawl" step as well.

Digging Deeper: This is sometimes referred to as the Minimal Viable Product - you can learn more about this process and adapt it into your project strategies by reading this article: [**Making Sense of MVP**](#) by Henrik Kniberg

Step 6: Input, Output, Store, Process: In this step students design the way their app will actually work to process data.

First students will draw and then describe the outputs of their apps. On the left side they can make a rough sketch of what their app would look like. This does not need to be a final draft and is just there to help them brainstorm ideas and communicate to another group how their app would look. On the right side they have space to label each individual piece of information on the screen.

Using the outputs that students selected as a guide, students should pick the inputs they'll need to create them. Only 6 spaces are provided though students could opt to choose more. This is somewhat intentional to help students scope the functionality of their app.

Students will describe the way their app processes data using as a model the way they would process it themselves. The goal here is primarily just to ensure students have selected inputs that could be processed to produce the outputs. For example, if they're finding a list of friends with birthdays this month then both a computer and human would need to know a list of friends' birthdays and the current month.

Lastly students are asked to decide what information, if any, it makes sense to store long term.

Distribute: Apps and Problem Solving - Peer Review, one copy to each pair of students

Step 7: Peer Review: Each group should trade their project guides with another. They should fill out the first line of the peer review which asks what specific part of their project they'd like feedback on. Afterward, there are a number of directed questions as well as a chance to provide more open-ended feedback on the idea.

Students should be given their project guides back as well as their peer feedback. On the back, there are questions where they can indicate what changes or improvements to their projects they intend to make on their apps based on the feedback.

Step 8: Finalize App and Make Poster: Students should incorporate the ideas of their peers in finalizing their app idea. They should then make a poster presenting their app following the guidance provided in the activity guide.

Students should revisit the rubric for this project to see how their project aligns. You may also decide to distribute the **Apps and Problem Solving - Student Checklist** so students can self-assess how well their project meets the requirements for this project.

Teaching Tip

Rubric and Checklist: Students have two resources they can use for self-reflection and making sure they are on the right track: the rubric and the student checklist. We recommend having students use the checklist for their own self-assessment and reflection, since it may be easier to digest and understand when reviewing their own project. However, we recommend teachers use the full rubric for evaluating projects to give more accurate feedback to students. You can see examples of this with the *Sample Marked Rubrics* resource at the top of the lesson plan (only visible to verified teachers)

Assessment Opportunity

Use the project rubric attached to this lesson to assess student mastery of the learning goals of this unit. You may also choose to assign the post-project test through code studio.

Presenting Apps

Share: Decide if and how students will share their posters with one another. If students will be doing more formal presentations then use the guidelines provided in Step 9 of the project guide to structure the presentations.

Collect: At the end of the presentation collect the completed project guides, peer feedback forms, and posters from each group.

Wrap Up (5 minutes)

Reflection

Question of the Day: How can the IOSP model help us to design an app that solves a problem?

Reflect: Students reflect on the following question in their journals:

- What are you most proud of in your project?

Send students to Code Studio to complete their reflection on their attitudes toward computer science. Although their answers are anonymous, the aggregated data will be available to you once at least five students have completed the survey.



End of Unit Survey

Teacher End-Of-Unit Survey

We also have a teacher end-of-unit survey to learn more about how the unit went for you and your students. While students take their survey, **please complete this end of unit survey for teachers** as well. Your feedback is valued and appreciated!

Extended Learning

Shark Tank

Run a mock "Shark Tank" as the backdrop for this unit project. Some things to consider:

- Is the culture of your class one where this competition can remain at a healthy level?
- Invite faculty / staff, local residents, or other professionals to hear the Phase 2 presentations and decide on a first, second, third place idea.
- Invite local business people / other professionals to share during Day 1 or Day 2 so students can see how computer science relates to jobs in their community.

Post-Project Test

Post-Project tests are included at the end of every unit. These include several multiple choice and matching questions as well as open ended reflections on the final project of the unit. These tests are aligned to the **learning framework** of each unit and are designed to assess parts of the framework that may not have been covered by the project rubrics. To holistically assess the learning objectives of the unit, the post-project test should be paired with the end-of-unit project which is the primary student assessment in each unit.

Unlocking The Tests: This test is locked and hidden from student view by default. In order for students to see and take this test, you'll need to unlock it by clicking the "Lock Settings" button and following the instructions that appear. **[Click here for more information about unlocking and administering assessments](#)**



This work is available under a **[Creative Commons License \(CC BY-NC-SA 4.0\)](#)**.

If you are interested in licensing Code.org materials for commercial purposes **[contact us](#)**.