

Lesson 10: Functions Practice

Overview

In this lesson students spend most of their time practicing using the skills and processes they have learned about functions. At the conclusion of the lesson students discuss remaining questions in anticipation of their Make project in the following lesson.

Purpose

This lesson is students primary opportunity to get hands on with functions in code prior to the Make activity in the following lesson. Give students as much class time as you can to work through these. For this lesson it's recommended that you place students in pairs as a support and to encourage discussion about the challenges or concepts they're seeing. In the following lesson students are encouraged to work independently.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- **AP** - Algorithms & Programming

Agenda

Lesson Modifications

Warm Up (5 minutes)

Quick Warm Up

Activity (35 minutes)

Practice Time

Wrap Up (5 minutes)

Assessment: Check For Understanding: AP Practice

Objectives

Students will be able to:

- Debug programs that use functions
- Identify opportunities to use functions to reduce repeated code within a program
- Write programs that use functions with the support of sample code

Preparation

- Review the video in the level progression that covers the topic of global vs. local variables
- Review the map level about "When to make a function?"
- Review other programming levels to be better prepared to support students

Links

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

For the teachers

- **CSP Unit 4 - Variables, Conditionals, and Functions** - Slides
- **Guide to Practice Lessons** - Video

For the students

- **Debugging Variable Scope: Functions** - Video

Teaching Guide

Lesson Modifications



Attention, teachers! If you are teaching virtually or in a socially-distanced classroom, please read the full lesson plan below, then click **here** to access the modifications.

Warm Up (5 minutes)

Quick Warm Up

Remarks

Today we're going to have a chance to practice programming with a lot of the concepts and patterns we've explored over the last two lessons. I encourage you to go through these with a partner, but pay close attention to what each other is doing. In our next lesson you're going to have to use a lot of these on an independent project, and these activities are good practice for what you'll find there! Alright, let's get to it!

Teaching Tip

Move Quickly to the Activity: There's a lot in the main activity of today's lesson. You may optionally wish to do a quick vocabulary review or address any questions that came up in the last lesson. Otherwise, give students more time to get hands on with some code.

Activity (35 minutes)

Practice Time

Teaching Tip

Providing Support: Circulate around the room through the lesson encouraging students to use the strategies introduced at the beginning of the lesson. Students have a number of supports at their fingertips, so a big part of your role is helping build their independence in using those resources.

Group: It is recommended that students work in pairs for this lesson and a number of the activities feature discussions. Consider using pair programming, having drivers and navigators switch every 3 minutes, not every level.

Distribute: Optionally pass out a plastic cup or other manipulative they can place on their computer when they are stuck as a signal that they need support.

 **Do This:** Direct students to Code Studio, Lesson 10 Level 2

Remarks

Group: It is recommended that students work in pairs for this lesson and a number of the activities feature discussions. Consider using pair programming, having drivers and navigators switch every 3 minutes, not every level.

Distribute: Optionally pass out a plastic cup or other manipulative they can place on their computer when they are stuck as a signal that they need support.

 **Do This:** Direct students to Code Studio, Lesson 10 Level 2

Today you're mostly going to practice what we've learned about programming with functions. I'm here to help you when you need, and you can put this cup on your computer when you need help. However, I first want to remind you of the following:

- Use your debugging skills. Try to zoom in on precisely where you're getting stuck.
- Talk to your partner! That's what they're there for!
- Hover over blocks to read the documentation about how they work.
- Read the resources in the Help & Tips tab
- Talk to the group next to you. If another group asks for help make sure to take some time to talk it through with them.

Levels 1 - 5 Declare and Call Functions: In these levels students practice declaring and calling functions. At first students practice calling functions that have already been declared for them. Students can focus their energy on the syntax of calling a function and how using functions changes the order in which lines of code run. Later in the progression they practice finding repeated code and are guided through how to create a function in its place.

💡 Teaching Tip

Providing Support: Circulate around the room through the lesson encouraging students to use the strategies introduced at the beginning of the lesson. Students have a number of supports at their fingertips, so a big part of your role is helping build their independence in using those resources.



1-5

Declare and Call Functions

1

2

3

4

5

Display: Scope Practice video. This video is located in the slides.

Levels 6 - 7 Function Scope: These two levels return to a topic that was covered in the variables lessons as well: variable scope. While students do not need a deep understanding of scope at this point, they will in some instances encounter debugging challenges that arise because of it.

💡 Teaching Tip

Level 7 is a Map Level, which contains a review of variable scope. It is highly recommended that you watch the video in the slides prior to the lesson. Slides are available for this lesson if you would like to optionally review as a class.



6-7

Function Scope

6

7

Level 8 - 9 Creating Functions: In these levels students revisit the Movie Ticket app and are challenged to think through the process of declaring a function with code that they anticipate could be repeated. Level 9 is a map level about when students should decide to create functions.

💡 Teaching Tip

Never - After - During - Before: In the map level students are introduced to a framework for thinking about their development with using functions. The primary question is "when should I make a function". In general, you want students to make their functions earlier, as this both improves the process of writing code and reflects deeper understanding of code structure. Throughout this course, you want to see them moving along the "never - after - during - before" scale.

At this point students are still learning to identify repeated code and replace it with a function. This would align with the "after" level. In the Functions Make project and the Unit 4 project they are encouraged to anticipate the need for functions in advance. Rather than write code twice and then remove the duplicate code by creating a function later, they should begin deciding early that they'll need a function. The "updateScreen()" pattern" helps reinforce this point.

Reinforce this language in the classroom, though remember it's only a guide. Not every student will immediately be able to move on to "during" or "before". Different approaches also work better in different contexts, and many experienced programmers will typically operate in the "during" mode unless they're building a large and complex project.



8-9

When to Declare Functions

8

9

Wrap Up (5 minutes)

Discuss: *What aspects of working with functions clicked today? What do you still feel like you have trouble with?*

Discussion Goal: Use this opportunity to address any lingering questions or misconceptions in the room. You can also use this as a source of discussion topics to kick off the following lesson. As you lead the discussion, call out the many resources students have access to help when they're getting stuck.

Assessment: Check For Understanding: AP Practice

Check For Understanding Question(s) and solutions can be found in each lesson on Code Studio. These questions can be used for an exit ticket.

Question: What will be displayed after this code segment is run?

day_of_week = "Saturday"

day_of_week = "Monday"

```
IF (day_of_week = "Saturday" OR day_of_week = "Sunday")
    weekend
ELSE
    weekday
```

```
PROCEDURE weekday
    DISPLAY "School day"
```

```
PROCEDURE weekend
    DISPLAY "Day off"
```

