

LESSON: Functions, parameters, and local variables - part 1		Time: 50 minutes
<p><b>Overview:</b></p> <p>This lesson first reviews functions and when they might be created. This is done by looking at previous programs and where functions could be created. All the examples are functions without parameters. Then an example is given that requires a parameter. Using two different programs, students learn about parameters and local variables. Then students practice recognizing when a variable is a parameter or local variable. <b>NOTE:</b> This lesson is continued (part 2)</p>		<p><b>Objectives:</b></p> <ul style="list-style-type: none"> <li>I can define “parameter,” “argument” and “local variable”</li> <li>I can determine when a variable should be a parameter or a local variable</li> <li>I can look at code and identify parameters and local variables</li> <li>I can write a function call statement with an argument</li> </ul>
<p><b>Standards:</b></p> <p><b>2-CS-03</b> Systematically identify and fix problems with computing devices and their components.</p> <p><b>2-AP-13</b> Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.</p> <p><b>3A-AP-17</b> Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.</p>	<p><b>CSP Framework:</b></p> <p>Computational Thinking Practices:</p> <p>2.B Implement and apply an algorithm.</p> <p>3.B Use abstraction to manage complexity in a program.</p> <p>3.C Explain how abstraction manages complexity.</p> <p>4.C Identify and correct errors in algorithms and programs, including error discovery through testing.</p>	<p><b>Key Concepts:</b></p> <ul style="list-style-type: none"> <li>A <b>parameter</b> is a piece of information a function needs to complete its task.</li> <li>A function will often use <b>local variables</b> that are defined in the function and used only in the function to accomplish its task.</li> <li>If a function calculates a value that is needed in the main program, it can be passed back, or <b>returned</b>.</li> <li>If a function has a parameter, the function call will pass a value to the parameter, called an <b>argument</b>.</li> </ul>
<p><b>Preparation:</b></p> <p><b>Decide</b> how you will have students work on the activity:</p> <ul style="list-style-type: none"> <li>Work individually or pairs on the assignment</li> <li>Work in groups at white boards</li> </ul>	<p><b>Links:</b></p> <ul style="list-style-type: none"> <li><a href="#">Instructions slide deck</a></li> <li><a href="#">Activity for printing</a></li> <li><a href="#">Assignment (individual)</a></li> <li><a href="#">Assignment (group work)</a></li> <li><a href="#">Activity Answers</a></li> <li><a href="#">Check Your Understanding</a></li> </ul>	<p><b>Agenda:</b></p> <ul style="list-style-type: none"> <li>Function review (10 minutes)</li> <li>Parameters &amp; local variables (5-10 minutes)</li> <li>Activity (30 minutes)</li> <li>Wrap-up &amp; Assessment (5 minutes)</li> </ul>
<p><b>Vocabulary: (review from Mission 9)</b></p> <ul style="list-style-type: none"> <li><b>Function:</b> a named set of instructions that accomplishes a task (A named chunk of code you can run anytime just by calling its name; also called a procedure)</li> <li><b>Parameter:</b> A local variable in a function that receives a value passed into the function when it is called; information the function needs to complete its task</li> <li><b>Argument:</b> The value passed into a function – information the function needs to complete its task. An argument can be a literal value, a variable, or an expression.</li> <li><b>Local Variable:</b> Variables defined inside a function, and can only be used within that function.</li> </ul>		
<p><b>Assessment:</b></p> <ul style="list-style-type: none"> <li>Daily reflection journal or Google form</li> <li>Assignment completion</li> <li>The concept is continued in the next lesson (part 2)</li> </ul>		

## Teaching Guide

✓ This lesson can be taught a couple different ways. Look at the options and decide what is best for your students. The lesson starts with warm-up questions and ends with a reflection. In between is the activity. Option 1 is for students to work in random groups at white boards. Option 2 is for students to work individually or in pairs with the assignment document. Both options will be presented in the teaching guide.

✓ NOTE: This lesson can be done without the assignment document if your students are working in groups at white boards. You can do the warm-up and reflection in a non-print way, or leave it out altogether. This is part 1, and the next day will continue the lesson with different problems.

### Preparation:

#### Option 1: Group Work

- Make a copy of **Assignment-Pt 1: Functions, parameters and local variables - board work** for each student in the LMS.
- Print the activity problems on paper for the groups to work through (**Activity Prints: Functions, parameters and local variables**).
- The **Check Your Understanding** could be printed and used for individual or pair work – an opportunity for students to try the problems on their own after the group work.
- Be familiar with the answers to the activity problems so you can support your students while they work.
- The **Check Your Understanding** is the same as the assignment, so you may not use it, or you can use it later in the week as a review.

#### Option 2: Individual or Pair work on assignment

- Make a copy of **Assignment-Pt1: Functions, parameters and local variables** for each student in the LMS.
- Be familiar with the answers to the activity problems so you can support your students while they work.

## Warm-up / Function Review (10 minutes)

The warm-up gives students an opportunity to review or see what they remember about functions.

### 💡 Teaching tip – warm-up

- Go through slides #2-3
- Students write their answers on their assignment. Discuss using a think-pair-share technique.

### 💡 Teaching tip – Review

- Go through slides #4-14 slides with the whole class
- This should be a review, recalling what they have already done in previous missions and remixes
- Spend the time you need to clear up any questions or misconceptions, but don't take too long; try to get to the new material within 10 minutes.

## New Concept (10 minutes)


### 💡 Teaching tip:


Go through slides #15-27. During the slides you will discuss:

- Parameters: how they are identified
- Parameters: how they are used in a function
- Function call with parameters
- Local variables: how they are identified
- Local variables: how they are used in a function

- One example also has a return statement – worth mentioning

## Activity (30 minutes)

 The key here is to have students identify ALL variables in the function code. Then decide if it is a parameter or a local variable.

-  Option 1: Group students randomly and have them work at vertical white boards
- Option 2: Students work individually or in pairs and go through the assignment document

### Teaching tip - Example:

Go over the example Slide #28 – this can be done with option 1 or option 2

### Teaching tip - 6 problems:

Mild and Medium problems

- Option 1: use the printouts of the problems and have groups of students work on the problems in any order and use class autonomy for students to check their work, discuss, etc. Use consolidation to end the activity.
- Option 2: students work through the six problems on their assignment document. You may stop the work periodically to share out answers and make sure students are understanding the concepts.

### Teaching tip - lesson continuation:


Reminder: this lesson is continued the next day (part 2). If you need to work (or rework) some of the problems again, students will have the opportunity in the next lesson. They will continue to learn about and practice functions, parameters and local variables.

## Wrap-Up (5 minutes)

The wrap-up is a review of functions, parameters and local variables. If students don't have time for the review, the lesson can be completed without it. If there is time, it is a good reflection for students, but not essential.

Formative Assessment:

- Daily reflection journal or Google form
- Class discussion on what they learned about abstraction and functions
- Assignment completion
- Check Your Understanding – can be homework or classwork. Answers in folder
- Exit ticket

 You decide what you want students to turn in for a grade. They may not turn in anything if they participated in group work throughout the class period. Or you can have them do the warm-up and reflection on the assignment document. Or they can do the Check Your Understanding. If students completed the assignment during the class period, they should turn it in.

### SUCCESS CRITERIA:

- Define parameter and local variable
- Explain reasons to use a parameter
- Explain reasons to use local variables
- Decide a function name for a section of code
- Determine parameters for a function
- Determine local variables for a function
- Write a function call