

# Lesson 2: Method Parameters

45 minutes

## Overview

**What are the similarities and differences between passing primitives and object references as parameters?**

Students revisit passing primitives and object references as parameters to identify similarities and differences between each. Students explore the impact of using an object reference as a parameter on the original object and identify scenarios when it is not ideal to use an object reference as a parameter. Students then learn about user stories and write user stories for their projects to identify required program features.

## Standards

Full Course Alignment

### CSA Conceptual Framework

- **MOD-2** - Programmers use code to represent a physical object or nonphysical concept, real or imagined, by defining a class based on the attributes and/or behaviors of the object or concept
- **VAR-1** - To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values

## Agenda

### Warm Up (10 minutes)

Parameterized Song

### Activity (30 minutes)

Object References as Parameters

Using Object References as Parameters

Writing User Stories

### Wrap Up (5 minutes)

Revisiting the Need to Knows

Assessment: Check for Understanding

AP Classroom Topic Questions

## Objectives

Students will be able to:

- Differentiate between passing primitive values and object references as parameters
- Write user stories to identify required program features

## Preparation

- Check the **Teacher's Lounge** for verified teachers on the CSA Forum to find additional strategies or resources shared by fellow teachers

## Links

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

For the students

- **U7L2 Extra Practice** - Handout

## Vocabulary

- **End User** - The person who will use the program
- **Pass by Value** - The process of making a copy of the actual value of a variable to pass to a constructor or method
- **User Story** - An informal explanation of a program feature written from the perspective of the user

## Teaching Guide


# Warm Up (10 minutes)

## Parameterized Song

### Remarks

We have used parameters in our constructors and methods to make our code more flexible and work with different inputs. Let's explore some of the benefits and differences in using primitives and objects as parameters in our programs.

**Group:** Place students in pairs.


 **Do This:** Click through the animated slide to explain the parameterized song problem. Assign pairs to solve the problem with primitives and with objects as parameters.

#### Teaching Tip

Try to have an equal number of pairs working with primitives as parameters and objects as parameters.

 **Do This:** Have students work with their partners to write pseudocode to outline their solution.

 **Do This:** Have pairs with the primitives as parameters compare their solutions with pairs with objects as parameters.

 **Discuss:** Click through the animated slide to display the prompts. Use the Retrieve-Pair-Share strategy to discuss the prompts.

- *What did you notice about each solution?*
- *What do you wonder about each solution?*

**Discussion Goal:** Students identify similarities and differences between passing primitives and passing object references as parameters. Students may wonder about the impact on original variables and objects when passed as parameters.

#### Teaching Tip


Students may have misconceptions about similarities and differences. Refer back to these throughout the lesson as they are addressed.


# Activity (30 minutes)

## Object References as Parameters (10 minutes)

### Remarks

Passing primitives and object references as parameters work differently, and we have to be careful when passing objects as parameters because of these key differences. Let's look at what these differences are and identify best practices for passing object references as parameters.

 **Do This:** Review the lesson objectives.

 **Do This:** Direct students to Level 1 on Code Studio to investigate the program with a partner. Students make the changes to the program as prompted.

 **Discuss:** Click through the animated slide to display the prompts.

- *What do you notice about the code in this program?*
- *What do you wonder about the code in this program?*




**Discussion Goal:** Students notice the method that uses an object reference as a parameter makes changes to the original object, while the method that uses a primitive value as a parameter does not make changes to the original variable. Students also notice that a method can only access the private data and methods of a parameter that is a reference to an object when the parameter is the same type as the method's enclosing class. Students may wonder why the method that uses an object reference makes changes to the original object while the method that uses a primitive value does not make changes to the original object. Students may also wonder if there are scenarios when modifying the original object is not ideal.

#### Teaching Tip

Ask additional guiding questions to help students recall aliases and how this occurs in this program. For example:

- *The formal parameter gets a copy of the value that is passed. When the actual parameter is an object reference, what is actually copied - the object itself or the object reference?*
- *When the formal parameter is initialized to a copy of the object reference, what is the relationship between the formal parameter and the original object reference?*
- *What is it called when two reference variables point to the same object?*


**Do This:** Click through the animated slides to demonstrate object references and primitives as parameters:

-  Object references as the actual parameter and the result in the formal parameter
-  Primitive values as the actual parameter
-  Accessing private data and methods of a parameter in a method

 **Discuss:** Click through the animated slide to display the prompts.

- *What does the `this` keyword do when it is used as a parameter?*
- *How is this similar to how we have used the `this` keyword before? How is it different?*

**Discussion Goal:** Students recall that the `this` keyword refers to the current object. Students identify that using the `this` keyword as a parameter passes the current object in the method call. Students suggest similarities and differences, such as using the `this` keyword to refer to the current object and using the `this` keyword to pass the entire object instead of only referring to its instance variables.

 **Do This:** Click through the animated slide to demonstrate using the `this` keyword as a method parameter.

## Using Object References as Parameters (10 minutes)

### Remarks

Passing object references as parameters to methods and constructors can cause problems if unintended changes are made to the original object. There may be times we do want this to happen, but there are also times when we do not want this to happen.

 **Discuss:** Click through the animated slide to display the prompts.


- *When is it ideal to pass objects as parameters? Why?*
- *When should we avoid passing objects as parameters? Why?*


**Discussion Goal:** Students suggest passing objects as parameters when a problem requires the original object to be modified by a method or constructor. Students also suggest avoiding passing objects as parameters when the method or constructor should not modify the original object.

#### Teaching Tip

If students struggle to identify ideal scenarios for when to pass objects as parameters and when to avoid doing so, ask additional guiding questions to help them identify potential issues that can occur. For example:

- *Are there times when we do want a method to modify an object? Consider the `Dessert` class we worked with for the Project Mercury Pastries Food Truck business. Are there times when we would want to modify the original `Dessert` object?*
- *What if we wanted to create and return a new `Dessert` object based on information from an original one?*

 **Do This:** Click through the animated slide to demonstrate that when an object reference is a constructor parameter, the instance variable should be initialized with a copy of the referenced object.

 **Do This:** Direct students to Level 2 on Code Studio to complete Levels 2 and 3. Students debug the program on Level 2, then continue to Level 3 to complete a choice level to use object references as parameters to methods.



2-3

## Using Object References as Parameters

2

3

## Writing User Stories (10 minutes)


### Remarks

As software engineers, one of our primary goals is to develop software that provides usefulness and value to users. Our goal for the Creative Coding Project is to create a program using The Theater that portrays a personal interest or solves a problem. When we develop programs to convey a personal interest, we typically have an audience in mind that we want to show our work to. When we develop programs to solve a problem, we typically have a specific type of user that we want to benefit from the program.

 **Do This:** Define *end user* and *user story*.

 **Display:** Show the video – *Software Engineering: User Stories*.

 **Do This:** Click through the animated slide to demonstrate example user stories.


 **Do This:** Have students write user stories for their project on page five of their Creative Coding with The Theater Project Planning Guide.


## Wrap Up (5 minutes)

## Revisiting the Need to Knows

### Remarks

We just learned a lot of new information today, which may have even answered some of the Need to Know questions you wrote down about the Creative Coding Project. As we progress through the unit, it is helpful to stop and note what we have learned that is related to or useful for the project.

 **Do This:** Have students review the questions they wrote in the "Need to Know" column on page two of their Creative Coding with The Theater Project Planning Guide. Students add new questions to this column, check off any answered questions, and write answers to any questions in the "Learned" column.

 **Do This:** Have students share what they added to their chart with a partner.

#### Teaching Tip

If time permits, you can also have students share as a class.

 **Do This:** Review the concepts covered in this lesson.

 **Display:** Key Vocabulary

---

## Assessment: Check for Understanding

*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.*

 4

 Check for Understanding

## AP Classroom Topic Questions

To assign questions from the AP Classroom Question Bank that align with this lesson, create a custom quiz in AP Classroom by searching the Question Bank for the Essential Knowledge statements listed at the top of this lesson plan. You can find instructions and video demonstrations to do this on **AP Central**.



This curriculum 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**.