

# Lesson 5: Overriding Methods

45 minutes

## Overview

### How can I override methods from other classes?

Students have overridden methods from the `Object` class and worked with class hierarchies that implemented overridden methods. Students explore writing and calling overridden methods and how the `super` keyword can be used to call the superclass version of a method. Students identify scenarios for overriding methods and practice writing and calling overridden methods using the console, The Neighborhood, or The Theater.

## Standards

Full Course Alignment

### CSA Conceptual Framework

- **MOD-3** - When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

## Agenda

### Warm Up (10 minutes)

#### Custom Notifications

### Activity (30 minutes)

#### Overriding Methods

#### Writing Overridden Methods

### Wrap Up (5 minutes)

#### Revisiting the Need to Knows

#### Assessment: Check for Understanding

#### AP Classroom Topic Questions

## Objectives

Students will be able to:

- Explain the purpose and functionality of overridden methods
- Write and call overridden methods

## Preparation

- Create code review groups if you are not reusing the same groups
- 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

- **U7L5 Extra Practice** - Handout

## Teaching Guide


### Warm Up (10 minutes)


# Custom Notifications

## *Remarks*

Some of the apps, programs, and devices we use have different ways to provide notifications that we can set based on our personal preferences. Let's explore how we might design our own app that has different ways to provide notifications to a user.

**Group:** Place students in pairs.

 **Do This:** Have students work with their partners to create a UML diagram to illustrate their solution. Have pairs share their solution with another pair, then have groups share as a class.


 **Do This:** Explain the potential solution.

## Activity (30 minutes)


### Overriding Methods (15 minutes)

## *Remarks*

This problem and the potential solution are good examples of why we might want to override methods. To provide specific implementations in our classes, we have overridden methods in the `Object` class, such as the `toString()` method and the `equals()` method. But we're not limited to overriding methods in the `Object` class! We can also override methods that we write ourselves.

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

**Group:** Place students in pairs.

 **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.

 1

### Investigate: Overriding Methods

 **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 that the program traverses an array of the parent class type, and each child class outputs different information when calling the method. Students also notice that the keyword `super` is used to call a superclass version of a method. Students may wonder why each class has a different version of the method used while traversing the array.

 **Do This:** Click through the animated slide to review overriding and review overriding `toString()` and `equals()`.

 **Discuss:** Use the Retrieve-Pair-Share strategy to discuss the prompt.

- *Why would we want to override methods?*
- *Where else have we seen the benefits of overriding methods?*


**Discussion Goal:** Students suggest that the parent class might have a generic version of the method, but each child class might need more specific versions of a method. Students realize that overriding makes it possible to create lists of the parent class type and call the method but get the overridden versions in each subclass. Students also make connections to polymorphism and recall how traversing an array of a superclass type allows them to easily call each subclass version of a method.

#### 🔗 Teaching Tip

Have students recall polymorphism to help identify some benefits of overriding methods.

 **Do This:** Click through the animated slide to review polymorphism.


 **Do This:** Click through the animated slide to explain and demonstrate rules for overriding methods.

 **Do This:** Click through the animated slide to explain and demonstrate calling a superclass method in a subclass using the `super` keyword.

## Writing Overridden Methods (15 minutes)

### 🎤 Remarks


We already know how to override methods from the `Object` class. Now let's practice overriding methods from a superclass and calling these methods!

 **Do This:** Direct students to complete Levels 2, 3, and 4 on Code Studio. Students complete a Check for Understanding on Level 2, then continue to Level 3 to debug code attempting to override a method. On Level 4, students complete a choice level to write an overridden method using the console, The Neighborhood, or The Theater.

☰ 2-4

Overriding Methods



 **Do This:** Click through the animated slide to have students participate in the Code Review Call and Response.

 **Do This:** Direct students to complete a code review on Level 5.

👥 5


Code Review: Overriding Methods


## 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.

---

## 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.*

 6

 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**.