

Lesson 2: Object Aliases and Equality

45 minutes

Overview

How can I check if two objects are equal?

Students discover the difference between comparing primitive values and object references. Students learn that two object references are considered aliases when they both reference the same object and how to identify aliases or `null` values using the `==` and `!=` operators. Students then revisit the `Object` class and override the `equals()` method to determine whether two objects of the class are equivalent.

Standards

Full Course Alignment

CSA Conceptual Framework

- **CON-1** - The way variables and operators are sequenced and combined in an expression determines the computed result.
- **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 (5 minutes)

Object Equality

Activity (35 minutes)

Aliases

Revisiting the Object Class

Override the `equals()` Method

Wrap Up (5 minutes)

But, Because, So

Assessment: Check for Understanding

AP Classroom Topic Questions

Objectives

Students will be able to:

- Explain the difference between `==` and `equals()`
- Write an `equals()` method to compare object data for equality

Preparation

- Gather sticky notes or scrap piece of papers (one for each student)
- 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

- **U4L2 Extra Practice** - Handout

Vocabulary

- **alias** - a reference variable that points to the same object as another reference variable


Teaching Guide

Warm Up (5 minutes)

Object Equality

Remarks

When we compare two primitive values to check for equality, we use the equality operator to check if both values are the same. It's easy to tell if two numbers or `boolean` values are the same, such as three equals three or false does not equal true. However, it's not quite as easy when comparing two objects for equality.

 **Discuss:** Click through the animated slide to display the prompts. Use the Hold That Thought strategy to discuss the prompts.

- *What does it mean to you for two objects to be considered equal?*
- *Do we consider these two objects to be equal? Why or why not?*
- *How about different names for the same object? Do we consider these to be equal? Why or why not?*

Discussion Goal: Students suggest that two objects are considered equal if they have the same values for their attributes. Students share whether they consider the example objects equivalent and provide their reasoning. For example, students may suggest that they are equivalent because they share the same values for their attributes or that they are not equivalent because they are two separate objects. Students also share whether they consider different names for the same object as equivalent and provide their reasoning. For example, students may suggest that they are equivalent because they both refer to one object or are not equivalent because they are two different references.

Teaching Tip

Share additional examples and ask students how they decide these are equivalent. For example,


- *If we had two basketballs, would they be considered equal? Why or why not?*
- *If we had a song on a playlist twice, would they be considered equal? Why or why not?*
- *Do any of you have a nickname? Are your nickname and your full first name the same object?*

Activity (35 minutes)


Aliases (10 minutes)

Remarks

A reference variable stores a pointer to an object in memory and does not store the object itself. Because of this, comparing reference variables for equality does not always give us the results we might expect.

 **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.


 **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 some object comparisons returned `true` while others returned `false`. Students realize that the reference variables assigned to another reference variable return `true`, but the reference variables pointing to new objects return `false`. Students may wonder why these comparisons return `false` when the values for the instance variables are the same and how they might check these values for equality.


Remarks

When reference variables point to the same object, they are known as aliases. These reference variables point to the same location in memory and not separate objects.

 **Display:** Show the video – *Aliasing*.

 **Do This:** Click through the animated slide to define *alias* and demonstrate the example.

Revisiting the Object Class (10 minutes)

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


- *When would it be useful to use the `==` or `!=` operators with objects?*
- *When would it not be useful to use the `==` or `!=` operators with objects?*

Discussion Goal: Students suggest using the equality operators to check if two reference variables are aliases or if an object is `null`. Students note that the equality operators are not useful when checking if two objects have the same values for the instance variables.

Remarks

Aliases are useful for checking if two reference variables point to the same object. However, sometimes we need to check if two objects are equal. There is a way we can do this in Java.

Group: Place students in pairs.

 **Do This:** Direct students to Level 2 on Code Studio to investigate the program with a partner. Students make the changes to the program as prompted.


 2

Investigate: Object Equality

Override the equals() Method (15 minutes)

Remarks

We previously learned that the `Object` class is the superclass of all classes in Java. It has methods that we can override to be more specific to our programs.

 **Do This:** Click through the animated slide to review the `Object` class and introduce the `equals()` method.

Teaching Tip

Have students recall the `toString()` method and overriding. Ask students how and why they implemented overridden versions of the `toString()` method in the `Dessert` classes and how they might override the `equals()` method in the `Dessert` classes.

Display: Show the video - *The equals() Method*.

Do This: Click through the animated slides to explain comparing `String` objects for equality:

- Using the `==` operator to compare two string literals
- How Java stores string literals in the Java String Pool
- Comparing using the `==` operator to compare string literals and `String` objects
- Comparing `String` objects for equality using the `equals()` method

Discuss: Click through the animated slide to display the prompts. Use the Hold That Thought strategy to discuss the prompts.

- How could we use the `equals()` method to check if these two `Dessert` objects are equal?
- What if the other `Dessert` object is `null`?

Discussion Goal: Students suggest overriding the `equals()` method to use `if` statements to check if the values for the instance variables are the same to determine if the two objects are equal. If the values are the same, the method should return `true`. Otherwise, it would return `false`. Students note that the method should first check if the other `Dessert` object is `null` before checking the values of the instance variables.

Teaching Tip

Ask students if there are instances where they would not want to compare all instance variables to determine if two objects are equal. For example,

- If we had two `Dog` objects that had instance variables for `name` and `breed`, would we want to check if the names are the same to determine if the two `Dog` objects are equal?

Do This: Direct students to Level 3 on Code Studio to complete Levels 3 and 4. Students complete a Check for Understanding on Level 3, then complete a choice level on Level 4 to override and use the `equals()` method.

3-4

Overriding the equals() Method



Wrap Up (5 minutes)

But, Because, So

Distribute: Give each student a sticky note or scrap piece of paper.

Do This: Have students respond to the prompt on the sticky note or scrap piece of paper.

Teaching Tip

If time permits, have students share their responses with a neighbor.

 **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.



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**.

The following Topic Questions in AP Classroom can be assigned as a formative assessment for this lesson:

- Topic Questions 3.7

Note: *Some Learning Objectives and Essential Knowledge statements in the suggested Topic Questions are covered in later units.*



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**.