

Lesson 3: Overloading Methods

45 minutes

Overview

How can overloading a method be useful in my programs?

Students revisit overloading and explore overloading methods to determine their impacts on the functionality of a program. Students practice writing overloaded methods to solve problems using the console, The Neighborhood, or The Theater before beginning their initial plans for their Creative Coding with The Theater Project.

Standards

Full Course Alignment

CSA Conceptual Framework

- **MOD-1** - Some objects or concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence

Agenda

Warm Up (10 minutes)

Coding Club Usernames

Activity (30 minutes)

Overloading Methods

Planning Program Structure

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 overloaded methods
- Write and call overloaded methods

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

- **U7L3 Extra Practice** - Handout

Teaching Guide

Warm Up (10 minutes)

Coding Club Usernames


Remarks


Your school's online coding club has given everyone a username that consists of their first name, followed by a dot, followed by their last name. For example, Grace Hopper's username would be Grace.Hopper. However, a software engineer accidentally deleted the data containing everyone's first and last name, leaving only the usernames of the club members.

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

- *How can we retrieve a club member's first and last name from their username?*

Discussion Goal: Students suggest using `indexOf()` to find the location of the dot. Students note that they can then use the two-parameter version of `substring()` to find the first name and then use the one-parameter version of `substring()` to find the last name.

 **Do This:** Discuss the possible solution.

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

- *What is the difference between the two versions of `substring()`?*
- *How does Java know which one to use?*
- *Why do you think the designers of Java decided to give the two versions the same name?*


Discussion Goal: Students recall that the one-parameter `substring()` only specifies where to start and goes to the end, whereas the two-parameter version can specify both the start and end. Students identify that Java can differentiate between the two by the method signature. Students suggest that the methods have the same name since they perform the same task but just give the user different options on the information required to perform the task.

Activity (30 minutes)


Overloading Methods (15 minutes)

Remarks

The `substring()` method is an example of an overloaded method. Let's take a look at another example.


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

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

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

- *Why did we use overloading with constructors?*
- *How might overloading be helpful with methods?*

Discussion Goal: Students recall that orders can vary with the Project Mercury Pastries food truck. Some customers want to order just a standard dessert, while other customers want to customize their dessert. Students suggest that overloading methods make code more flexible for program requirements.

 **Do This:** Direct students to Level 1 on Code Studio to predict the program's outcome, then run the program to compare their predictions to the actual outcome.

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

- *What is the difference between each version of the overloaded method?*
- *Why do you think the different methods exist?*

Discussion Goal: Students note that the different versions of the methods take in different combinations of parameters. Students share that the methods all perform the same task but give the user different options for performing the task. Students realize that overloaded methods can also call another version of the method by passing in default values to keep things DRY (Don't Repeat Yourself).

Do This: Explain that methods are said to be overloaded when there are multiple methods with the same name but a different signature.

Do This: Direct students to complete Levels 2 and 3 on Code Studio. Students complete a choice level by writing overloaded methods using the console, The Neighborhood, or The Theater. Students continue to complete a Check for Understanding on Level 3.



2-3

Writing Overloaded Methods

2

3

Planning Program Structure (15 minutes)

Remarks

You did some brainstorming for your Creative Coding Project in a previous lesson. Let's now take some time to create an initial plan to guide your development processes in upcoming lessons. Your planning might include outlines, rough drafts, flowcharts, UML diagrams, or pseudocode for key algorithms in your program. Take a look at your Project Planning Board to decide what task you need to tackle first.

Do This: Review the first benchmark.

Do This: Have students work on their Creative Coding with The Theater Project.

Teaching Tip

Students might be unsure what task they should complete next. Encourage students to focus on brainstorming and planning tasks to identify the personal interest or problem they want to portray or address in their program and identify the major components that they need to develop. Refer students to pages six through eleven for planning space and templates.


Do This: Have students update their Project Planning Board and Project Backlog with any tasks they completed, changed, or added.


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 the 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.

 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**.