

Lesson 2: Integer and Double Objects

45 minutes

Overview

Why would I want to represent primitive values as objects?

Students construct `Integer` and `Double` objects to represent `int` and `double` primitive values. Using the methods in the `Integer` and `Double` classes, students return the values these objects store as primitive values. Additionally, students explore the minimum and maximum values for integers to avoid overflow errors.

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.
- **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 (5 minutes)

Primitive and Reference Types

Activity (35 minutes)

Exploring Wrapper Classes

Wrapper Class Methods

Using Wrapper Classes

Wrap Up (5 minutes)

But, Because, So

Assessment: Check for Understanding

AP Classroom Topic Questions

Objectives

Students will be able to:

- Describe the result of exceeding the maximum or falling below the minimum values for an integer
- Represent `int` and `double` primitive values as `Integer` and `Double` objects

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

- **U6L2 Extra Practice** - Handout

Vocabulary

- **Autoboxing** - The process of automatically converting a primitive type value into an object of the corresponding wrapper class
- **Overflow Error** - An error that occurs when an operation makes an integer value greater than its maximum
- **Parsing** - The process of dividing text into parts for analysis or conversion

- **Unboxing** - Automatically converting an object of a wrapper class to its corresponding primitive type
- **Underflow Error** - An error that occurs when an operation makes an integer value less than its minimum
- **Wrapper Class** - A class used to convert primitive data types into objects

Teaching Guide

Warm Up (5 minutes)

Primitive and Reference Types


Remarks

We have worked with both primitive and reference types to store primitive values and objects. Let's review some of their differences.

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

- *What are some differences between primitive and reference types?*

Discussion Goal: Students recall differences between primitive and reference types, which may include how they are declared and initialized, stored and accessed in memory, and composed of data types internally and contain methods.


 **Do This:** Click through the animated slide to review the differences between primitive and reference types.

Activity (35 minutes)


Exploring Wrapper Classes (10 minutes)

Remarks

We have worked with integer and decimal values as primitive data types. Java also allows us to create objects for `int` and `double` values. Let's take a look at how this works.

 **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 `Integer` and `Double` objects are instantiated to store integer and decimal values. Students also notice that `Integer` and `Double` objects can be created using both the `new` keyword and the assignment operator. Students may wonder why they would use `Integer` and `Double` objects instead of `int` and `double` values.

Teaching Tip

Ask students to consider the similarities and differences between creating `Integer` and `Double` objects and `String` objects. Have students make predictions about what might be happening when `Integer` and `Double` objects are created without the `new` keyword.

Display: Show the video – *Wrapper Classes*.

Wrapper Class Methods (10 minutes)

Remarks

The `Integer` and `Double` classes have methods to work with integer and decimal values. Let's take a look at some of their methods and capabilities.

Do This: Click through the animated slide to demonstrate the `Integer` class and its methods.

Do This: Click through the animated slide to demonstrate the `Double` class and its methods.

Do This: Click through the animated slide to demonstrate the `Integer` and `Double` constructors.

Do This: Define *parsing*.

Remarks

When you were exploring the program, you might have noticed something interesting that happens when you add one to the maximum value for an integer or subtract one from the minimum value for an integer.

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 the maximum and minimum values for an `int`?*
- *What happens when you add or subtract one from the maximum or minimum values?*

Discussion Goal: Students share the maximum and minimum values and their findings from adding to the maximum value or subtracting from the minimum value.

Do This: Click through the animated slide to explain the minimum and maximum values for integers.

Do This: Click through the animated slide to define *overflow* and *underflow*.


Using Wrapper Classes (15 minutes)

Remarks

Representing integer and decimal values as objects are useful for specific scenarios. When represented as objects, we can use the methods in the `Integer` and `Double` classes to work with these values.

 **Do This:** Click through the animated slide to demonstrate autoboxing.

 **Do This:** Click through the animated slide to demonstrate unboxing.

 **Do This:** Direct students to Level 2 on Code Studio to complete Levels 2, 3, and 4. Students debug the program on Level 2, then continue to complete Levels 3 and 4 to use `Integer` and `Double` objects and their methods.



2-4

Using Wrapper Classes


2

3

4

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.

 5



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**.

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

- Topic Questions 2.8

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**.