

# Lesson 10: Two-Way Selection Statements

45 minutes

## Overview

How can two-way selection statements and the NOT ( `!` ) logical operator improve my algorithms?

Students expand their knowledge of selection statements and `Boolean` expressions using two-way selection statements and the NOT ( `!` ) logical operator. Students practice using these to check the state of an object and executing a specific set of instructions based on the result.

## 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.
- **CON-2** - Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values

## Agenda

### Warm Up (5 minutes)

CS in Fashion

### Activity (30 minutes)

Two-Way Selection Statements  
Writing New Algorithms

### Wrap Up (10 minutes)

Revisiting CS in Fashion  
Assessment: Check for Understanding  
AP Classroom Topic Questions

## Objectives

Students will be able to:

- Explain the purpose and functionality of an `if-else` statement
- Use the NOT ( `!` ) logical operator to execute specific blocks of code if a condition is not `true`
- Write `Boolean` statements that execute specific blocks of code based on the outcome of a condition

## Preparation

- Create code review groups if you are not reusing the same groups from the previous lesson
- Print copies of the Using Two-Way Selection and NOT handout (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

- **Two-Way Selection and NOT** - Video
- **U1L10 Extra Practice** - Handout

- **Using Two-Way Selection and NOT** - Handout

## Vocabulary

- **NOT ( ! ) operator** - a logical operator that returns `true` when the operand is `false` and returns `false` when the operand is `true`
- **if-else statement (two-way selection statement)** - specifies a block of code to execute when the condition is `true` and a block of code to execute when the condition is `false`
- **logical operator** - an operator that returns a `Boolean` value

## Teaching Guide


### Warm Up (5 minutes)

CS in Fashion

#### Remarks

Computer science influences industries that we might not expect, such as the fashion industry.

 **Do This:** Share the example of computer science in the fashion industry.

 **Do This:** Direct students to write down ideas and thoughts in response to the prompt on a sticky note or scrap piece of paper.

 Teaching Tip


Students participate in a Give One, Get One activity during the wrap up to share their ideas with their peers.

### Activity (30 minutes)

Two-Way Selection Statements (15 minutes)

#### Remarks

We have learned to use `if` statements to execute a block of code if a condition is `true` and `while` loops to repeat a block of code as long as a condition is `true`. What if we wanted to execute a specific block of code if a condition is NOT `true`?

 **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: Selection and Logic

**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 note that the program uses `if-else` statements to specify a block of code to execute if the condition is `true` and a block of code to execute if the condition is `false`. Students identify the `NOT ( ! )` logical operator and may determine its purpose. Students may wonder how `true` statements and the `NOT ( ! )` logical operator can be combined with `while` loops.

**Display:** Show the video - *Two-Way Selection and NOT*.

## Writing New Algorithms (15 minutes)

### Remarks

We have learned to use pseudocode to plan our algorithms before implementing them in Java. We get better at planning algorithms the more we practice!

**Do This:** Click through the animated slide to demonstrate writing pseudocode for an algorithm using an `if-else` statement and the `NOT ( ! )` logical operator.

**Group:** Place students in pairs or small groups of three.

**Distribute:** Give each student a copy of the Using Two-Way Selection and NOT handout.

**Do This:** Direct students to write pseudocode for the methods and add them to the `PainterPlus` UML diagram.

**Do This:** Direct students to Level 2 on Code Studio to complete Levels 2, 3, and 4. Students complete a Check for Understanding on Level 2, then continue to complete Levels 3 and 4 to implement their algorithms in the `PainterPlus` class.

 2-4

## Using Two-Way Selection and NOT




### Remarks

This is a good time to commit our code and save our new version of the `PainterPlus` class to the Backpack. Anytime we make changes to our programs, it is helpful to commit, or save, our work as a new version in case we need to revert to a previous version.

**Do This:** Play the music clip to cue committing their code and saving the new version of the `PainterPlus` class to the Backpack.

### Remarks

When we write new code, getting feedback from our peers is helpful to make sure we have met the requirements of the problem efficiently.

 **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: New PainterPlus Methods

## Wrap Up (10 minutes)

### Revisiting CS in Fashion

#### Remarks

At the beginning of this lesson, we heard about how computer science is used in the fashion industry. Let's revisit the ideas and thoughts you had and share these with your peers. It's time to...

 **Do This:** Play the music clip to cue the Give One, Get One activity and direct students to participate in a Give One, Get One.

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

- *What did you learn from each other?*
- *How do these ideas and thoughts contribute to your identity as a software engineer?*

**Discussion Goal:** Students share the ideas they gathered and identify patterns or common thoughts. Students realize that computer science influences industries outside of developing technology and make personal connections to their identity as software engineers.

 **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. You can use these questions as 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**.

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

- Topic Questions 3.3

**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**.