

# Lesson 2: Variables Investigate

## Overview

In this lesson students work with partners to investigate several versions of the "Thermostat App" to understand how variables store and update information. To begin, students examine a version of the app where the temperature displayed changes each time a button is clicked. The next two versions of the app demonstrate how variables can store strings. Students learn about the patterns they are observing, specifically "Counter Pattern with Event" and "Variables with String Concatenation Pattern". To conclude the lesson, students review and discuss the programming patterns that they will make use of in the programs they write.

## Purpose

After building a conceptual model for variables in the previous lesson, students investigate three working examples of apps that make use of variables. This lesson also introduces common programming patterns when using variables. Students will have some opportunities to modify working code in this lesson, but the most significant practice with variables will come in the following lesson.

## Standards

Full Course Alignment

### CSP Conceptual Framework

- **AAP-1** - To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.
- **AAP-2** - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.

### CSTA K-12 Computer Science Standards (2017)

- **AP** - Algorithms & Programming

## Agenda

### Lesson Modifications

### Set Up

## Objectives

Students will be able to:

- Explain the purpose of those programming patterns with variables both in terms of how they work and what they accomplish
- Identify common programming patterns when using variables as part of an app
- Modify apps that make use of common programming patterns with variables to adjust their functionality

## Preparation

- Review the example apps and the prompts that students will be asked to respond to for each
- Review the information covered in the slides

## Links

**Heads Up!** Please make a copy of any documents you plan to share with students.

For the teachers

- **CSP Unit 4 - Variables, Conditionals, and Functions** - Slides
- **EIPM: A Short Introduction** - Resource
- **Guide to EIPM Lessons** - Video
- **Guide to Investigate Lessons** - Video

**Warm Up (5 minutes)**

**Preview the Lesson**

**Activity (35 minutes)**

**Investigation #1: Thermostat App v.1 (10 minutes)**

**Investigation #2: Thermostat App v.2 (10 minutes)**

**Investigation #3: Thermostat App v.3 (8 minutes)**

**Patterns (7 minutes)**

**Wrap Up (5 minutes)**

**Assessment: Check For Understanding**

## Teaching Guide

## Lesson Modifications



**Attention, teachers!** If you are teaching virtually or in a socially-distanced classroom, please read the full lesson plan below, then click **here** to access the modifications.

## Set Up

This is the first official "Investigate" lesson in the EIPM model. Review the EIPM model in the **EIPM One Pager**.

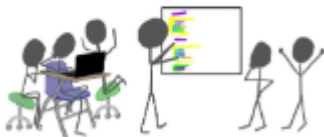
### Investigate Lessons:

**Overview:** Students investigate two or three sample programs that use the new concept.

- Close-reading of working programs
- Teacher-led discussions
- Tasks to modify apps

**Goal:** Students become comfortable reading and modifying programs that use the new concept.

## Investigate



Watch the following videos:

- **Guide to EIPM Lessons**
- **Guide to Investigate Lessons**

## Warm Up (5 minutes)

## Preview the Lesson

### Remarks

Yesterday we explored storing information like a computer. Computers store each piece of information in a variable. In Javascript, we name or declare a variable with the keyword `var`. Today we are going to look at a new app that stores information in variables.


 **Discuss:** *Let's do a quick review. How does a baggy represent a variable?*

**Discussion Goal:** The baggy represents a variable by storing one item: a value on a sticky note placed in a named baggy. The value in the baggy can be changed at any time, just like a variable's value can change.

## Activity (35 minutes)

 **Group:** Place students in pairs. One student per group should navigate to the lesson on Code Studio.

### Investigation #1: Thermostat App v.1 (10 minutes)

 **Level 1: Thermostat App v.1:** This level introduces a new app for students to investigate. It represents a Thermostat App where the temperature can be changed up and down.

- Run the app: Let students run the app for a few minutes.
- Predict: Students predict the information that is being stored in variables.

 **Level 2: Thermostat App v.1 Code:**

- Assign Code Sections: Assign half the pairs to investigate the first section on lines 1-12. The other half investigates the second half on lines 14-21.
- Read Code: Groups should carefully read the code for their section making sure they understand how it works. Give them 3 minutes to do so.
- Explain Your Section: Have partners make a group with members of the other section and carefully explain how their section works line by line.

 1-2

Investigation #1: Thermostat App v.1

1

2

**Discuss:** Ask a few members of each section to quickly share out how their section works. Display the code at the front so you can talk through it together.

Here are some points that students are likely to bring up while discussing their code:


**Lines 1-12:**


- Two variables are created: `temp` and `tempF`. `temp` gets the value `70` and `TempF` is set to an empty string, which is represented with empty quotation marks.
- On line 7, an `onEvent` is created for when the `downButton` is clicked.
- When the down button is clicked, the `temp` variable decreases by one.
- Then the `tempF` variable is updated with the current value of `temp` joined with the letter `F`.
- The text on the screen is set to display the value stored in the variable `tempF`.

- A sound is played.


#### Lines 14-21

- On line 16, an `onEvent` is created for when the `upButton` is clicked.
- When the up button is clicked, the `temp` variable is increased by one.
- Then the `tempF` variable is updated with the current value of `temp` joined with the letter `F`.
- The text on the screen is set to display the value stored in the variable `tempF`
- A sound is played

 **Display:** Display the slide showing students how to add a watcher in the debugging panel to track the value a variable stores.

 **Modify:** have groups return to their original seats. Give them a couple of minutes to work on modifying the app to change the degrees by two when the up and down arrows are clicked.

## Investigation #2: Thermostat App v.2 (10 minutes)

 **Level 3:** This program is an updated version of the Thermostat app. This time students should continue to work in partners but do not need to work with other groups. They will need to:

- Run the app: Let students run the app for a few minutes.
- Discuss Changes: Talk through how the app is different than the first version.
- Find the `Math.round` command: Discuss with a partner how this command might work. Try deleting `Math.round` in lines 3, 15, and 28. What happens? Add it back in.

 3

### Investigation #2: Thermostat App v.2

- Class Discussion: Ask a few students to explain what's happening on line 3 with `Math.round`.
- Modify the Code: Change the code so that no space displays between the temperature and the unit description ("F" or "C").

**Discussion Goal:** On line 3, the Fahrenheit temperature is converted to Celsius. `Math.round` rounds a given value to the nearest integer.

## Investigation #3: Thermostat App v.3 (8 minutes)

 **Level 4:** This program is again an updated version of the Thermostat app with a login screen.

- Run the app: Let students run the app for a few minutes.
- Predict: Students predict the information that is being stored in variables.

 **Level 5:** The code for Thermostat App v.3 is displayed.

- Read Code: Give students a few minutes to read and discuss the code with their partners. What has changed? What has stayed the same?

 4-5

### Investigation #3: Thermostat App v.3

4

5

- Class Discussion: Look at line 40. How does `getText()` work? What is it doing?

- Modify the Code: Add an explanation point `"!"` to the end of the string stored in `userName`.

**Discussion Goal:** `getText` gets the string of text that is found in the element with the given id. In this case, the id is `"nameInput"`. In our app, the name written in the `"nameInput"` element is concatenated with `"Hi"` and then stored in the variable `myName`.


## Patterns (7 minutes)

### *Remarks*

We are learning two patterns here:

- Counter Pattern with Event
- Variables with String Concatenation Pattern

The Counter Pattern with Event is a common pattern for updating variables that you will use in making many different apps. We call it the **Counter Pattern with Event**.

 **Display:** Talk through the pattern with students.

**Discuss:** *With a partner, discuss the following:*

**Discussion Goal:** Students should step through each line of the pattern, explaining what's happening.

Counter Pattern with Event:

- `myVar` gets the value 0
- When the event is triggered, `myVar` is updated getting the current value of `myVar` and adding 1.

The **Counter Pattern with Event** might be used to update a score in a game when an item is clicked.

- The "Counter Pattern with Event" should look familiar! How would you explain this pattern to another person?
- When might you want to use the "Counter Pattern with Event"?

### *Remarks*

Variables can store many different types of information including numbers and Strings. Anything placed inside of the quotation marks becomes a String.

Guess what! There's a pattern for working with Strings. When we want to combine Strings with other Strings, numbers, or even with another variable we call that concatenation.

 **Display:** Talk through the pattern with students.

**Discuss:** *Explain to a partner how the "Variable with String Concatenation Pattern" works.*


**Discussion Goal:** Students should be able to verbally explain how the **Variable with String Concatenation Pattern** works.

- `myString` gets the value `"rock"`
- `myOtherStrings` gets the value `"roll"`
- `myStory` gets the value that is stored in `myString` (`"rock"`) and combines that with a string `"and"` and the value stored in `myOtherString` (`"roll"`)
- `myStory` now stores the value: `"rock and roll"`

## Wrap Up (5 minutes)

### Remarks

The patterns we learned today are present in many different kinds of apps. As we learn new patterns, they will show up in the "Help & Tips" tab at the top of your screen on programming levels.

 **Discuss:** *Let's review. What can be stored in a variable? Why is using a meaningful name for the variable important?*

#### Discussion Goal:

What can be stored in a variable?

- At this point, students should know that these two things can be stored in variables. Students will learn about other data types that can be stored in variables in later lessons.
  - Numbers
  - Strings
    - Including concatenated strings ( `temp + " F"` )

Why is using a meaningful name for the variable important?

- When you write code, you are not just writing for the computer, you are also writing for people. Using meaningful names help people be able to read your code easier.
- It also helps readers predict what values may be stored in the variables.

 **Journal:** Add to your journal the definition for variable.

### Remarks

Today we've learned a lot about how variables work in actual programs.

## 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.*

**Question:** Explain in your own words the process of creating and updating a variable. How does the Counter Pattern with Event work?

 6

 Assessment: Check For Understanding