LESSON NAME:

Wrap-Up

Lesson time: 45–60 Minutes : 5–20 minutes

Main Goal: Prepare the class for continued learning in computer science.

OVERVIEW:
The final wrap-up in the first 20-hour series will help the class relive all of their favorite moments, while giving them next steps that they can take on their own if they are interested in continuing their computer science journey.

OBJECTIVE:
Students will —
- Recall events covered over the last 19 sessions
- Reinvent concepts already covered by creating new games
- Work in groups to blend seemingly unique subjects

MATERIALS:
This is up to the instructor. It's a good idea to have a large variety of items to pique the classroom interest and prompt clever inventions. Some inspiring materials tend to be:
- Battery-operated tea lights
- Aluminum foil
- Paper clips

- Markers
- Paper (lined, blank, graph, construction)
- Scissors
- Tape
- Fake coins/money
- Beads
- String
- Popsicle sticks
- Pipe cleaners

PREPARATION:
You may want to come up with some ideas beforehand to guide students toward, but you can also literally just pile the supplies on the table and begin!

VOCABULARY:
Review of previous lessons' vocabulary.

Call attention to the fact that you have almost completed the entire entry series for computer science. That is a huge achievement, given that they have learned more over the last 20 lessons than most adults ever know about the subject.

INTRODUCE:

Sometimes, a large quantity of knowledge can sneak in without a class realizing it! This is your opportunity to be explicit about all of the things that your class learned over the last 20 lessons.

Start out by calling attention to the fact that you have almost completed the entire entry series for computer science. That is a *huge* achievement, given that they have learned more over this last little while than most adults ever know about the subject. This puts them in an elite category of thinkers, and they should really consider "paying it forward."

Let's take a second to remember all of the things that we've done. What was our first lesson about? What was your favorite part about that? What did we learn next? What was your favorite lesson of all time? What was the most challenging? Do you think it would feel easier now if we were to do it over?

When all is said and done, practice is key. Computer Science is a skill just like any other, and things start to feel a lot more simple the more you practice.

Let's take a look at all of the lessons that we've done, and break them into groups:

1) *Intro:* What is Computer Science?
2) *Maze #1:* Sequence, Loops, Conditionals, Nesting
3) *Computational Thinking:* Decompose, Patterns, Abstraction, Algorithms
4) *Graph Paper Programming:* Draw what the algorithm tells you
5) *Artist #1:* Draw Shapes, Loops, Increment
6) *Algorithms:* Put shapes into pictures, Folding paper
7) *Artist #2:* Figure out algorithm
8) *Functional Activity:* Suncatchers - Program, functions, variables
9) *Farmer #1:* Conditionals, Repetition, Variables
10) *Conditionals Exercise:* Coding with Cards
11) *Artist #3:* Calling functions, Repeat with Loops, Variables & Parameters
12) *Song Writing:* Functions like a chorus, Passing parameters, Parameters as Variables
13) *Farmer #2:* Functions
14) *Abstraction:* Madlib style stories
15) *Artist #4:* Functions and Parameters
16) *Coding Under Pressure:* Double Checking, Debugging
17) *Farmer #3:* Importance of Order, Debug pre-made program
18) *Internet:* What is the Internet? How does it work?
19) *Artist #5:* Free play
20) *CS Wrap-U :* What did we learn? What was your favorite part?

Decide what size you want the students teams to be; teams of 2-5 tend to work best. Calculate the number of teams that you will need, then divide up the lessons among the teams. For example, if you have 5 teams, you could assign the lessons like this:
{Team 1: Lessons 1,6,11,16}, {Team 2: Lessons 2,7,12,17}, etc. …

Challenge each team to come up with a game or activity that incorporates skills from each of the lessons in their set. Give them free access to the materials table, and set them loose.

When you reconvene, have each group describe their game in 2-3 minutes. If they sound like they're onto something, you can set aside another day to let them think the activities through, and still another day to play the games that the teams created.

When all is said and done, congratulate the class on an amazing section. Computer science is a rare skill, and now your class has the basics of it!

Don't forget to give them access to more resources that they can use on their own to continue the computer science learning. Some of the best are highlighted at: **code.org/learn/beyond** and **csisfun.com**.

ADJUSTMENTS:

K-2: For young students, you may want to just spend the discussion asking them about their favorite games from this series, and the things that they remember most. Ask them about some of the most important vocabulary, like: Functions, Variables, or Debugging. If you have time, vote on which game they would like to play one more time, and see how much better they have gotten!

3-5: This lesson should go pretty well as written, but you may find it's easier if you give them just two ideas to combine into a game, rather than five or more.

6-8: Have at it! You should get some really interesting games out of this activity. Hopefully, your students will give great insight as to how much they really learned and digested.