Duck, Duck, Loops

Educator's Guide

Overview

CS Hands-On is a 501(c)(3) nonprofit teaching computational thinking skills through technology-free lessons and activities. This curriculum is built to teach fundamental computer science concepts in an engaging, hands-on way. In this mission, students use for loops to play a spin-off of the game Duck Duck Goose.

- ### Prerequisite Knowledge

  Student should have completed the Looping Dance Party and Wacky While Loops activities, which introduce the concept of loops and while loops.

- ### Lesson Details

  At Patteron, students will learn to find and distinguish different patterns with Pancho the Snail. Students will learn the structure and syntax of a for loop, then use for loops to play a game of Duck Duck Loops.

  This lesson was developed for students ages 8 to 13 and can be modified for all skills and ages. This lesson takes roughly 30 minutes.

Learning Objectives

- ### Key Question

  How can you create a for loop and use for loops to play Duck Duck Loops?

- ### Key Terms

  For Loop: A type of loop that repeats a set of instructions a specific number of times using a start, end, and increment value

  Definite Iteration: When the number of repetitions in a loop is already specified in advance

- ### Curriculum Standards

  Students should be able to...

  - Explain how for loops are created and used (Patterns)
  - Read, write, and interpret for loops (Literacy)
  - Act out the process of a for loop (Creative Arts)

  View standards addressed here

Lesson Plan

- **Materials**
  - Duck Duck Loops worksheet (per student)
  - A pair of dice (per group)
- **Setup**
  - Hand out a Duck Duck Loops worksheet to each student
  - Set up your classroom to have space for students to sit in a circle

ANSWER KEY & LESSON ANNOTATIONS

Patteron
Mission 3

Name: _____ Date: _____

Duck, Duck, Loops

Loop de Loop

Ready to loop around for another trip at Patteron? Today, Pancho will show you the fabulous fun of for loops!

What are For Loops?

A **for loop** is a type of loop that repeats a set of instructions a specific number of times. For loops have a **definite iteration**, which means the number of repetitions is specified in advance.

More specifically, for loops have a **start**, an **end**, and an **increment** value to keep track of how many times to repeat its instructions. An **increment** is a number we want to increase our count by.

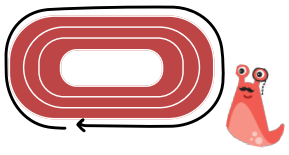
It may look like a bunch of wacky symbols at first glance, so follow along to see how we can break this down!

start ↓ end ↓ increment ↓
For count = 0, count == 5, count++:
Do a jumping jack

Reading this in English, the for loop uses a variable "count" to control the loop. The loop sets count equal to 0, stops counting when count equals 5 and increases the count by 1 once you have done a jumping jack.

Gym Class

Let's look at an example. Every day, Pancho loves to run around his school track. Pancho will be running 3 laps for his workout today, increasing his speed by 0.2mph each time. Here's how to represent this in a for loop:



First Pancho jots down the start, end, and increment:

Start = 0 laps (we're starting with 0 laps)
End = 3 laps (we want to count until 3 laps)
Increment = 1 lap (we're counting lap by lap)

CS Hands-On Duck, Duck, Loops 1

Educator Note

Explain to students the difference between the "=", "==", and "+=" symbols.

What is =? The single equal sign assigns a value to something. In "count = 0," we are setting 'count' equal to zero.

What is ==? In Logicland's Mixed Up Dots lesson, we learned that the "==" symbol checks if two things are equal to each other. In "count == 5," we are checking if the value count equals 5.

What is +=? We use the plus equals symbol to indicate that we are adding the value on the right to the value on the left. In "count+=1", we are adding 1 to the count value.

Now, we place this information into our for loop. We start our variable *count* at zero, stop counting when it reaches 3, and increase *count* by 1 each time we complete the instructions (increase speed by 0.2mph and run 1 lap).

For loop syntax:

start ↓ end ↓ increment ↓

For count = 0, count == 3, count++:
Increase speed by 0.2mph
Run 1 lap

⟶

What's happening:

Start -> count = 0
Increase speed by 0.2mph
Run 1 lap
count = 1
Increase speed by 0.2mph
Run 1 lap
count = 2
Increase speed by 0.2mph
Run 1 lap
count = 3 -> Stop!

Counting by miles

Surprise, Pancho's coach now has a new workout for him! Pancho will be running 4 miles, increasing his speed by 0.3 mph every mile. Instead of counting by laps, Pancho now wants to count by miles. 1 mile is 4 laps around the track.

Start = 3 laps (Pancho has run 3 laps so far)
End = 15 laps (we want to run a total of 4 miles and 3 laps -> 12+3 = 15 laps)
Increment = 4 laps (we're counting in increments of 1 mile = 4 laps)

For loop syntax:

start ↓ end ↓ increment ↓

For count = 3, count == 15, count+=4:
Increase speed by 0.3mph
Run 4 laps

⟶

What's happening:

Start -> count = 3
Increase speed by 0.3mph
Run 4 laps
count = 7
Increase speed by 0.3mph
Run 4 laps
count = 11
Increase speed by 0.3mph
Run 4 laps
count = 15 -> Stop!

Reflect

What are other scenarios where our increment value will be a number other than 1?
Examples include counting socks by pairs of socks, eggs by dozens of eggs, and cards by a deck of cards.

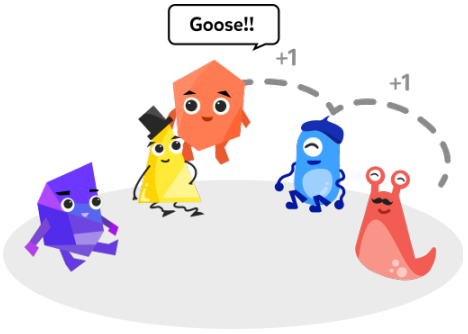
Duck, Duck, Loops

Materials

- A pair of dice

How to Play

- Sit in a circle with a group of 4-10 people.
- Choose one player to be the tapper. Every turn, the tapper will roll the pair of dice. The increment will be the 1st number and the end value will be the 1st number times the 2nd number.
- The tapper will walk around the circle, tapping players on the head based on the increment and saying "duck". Once they reach the end number, he/she says "goose" to make the player the goose. The goose then runs after the tapper, trying to tag them before the tapper takes their seat.
- If the tapper successfully reaches the goose's seat without being tagged, the goose is now the new tapper. However, if the goose tags the tapper, then the goose keeps their seat in the circle and the tapper continues tapping.



Educator Note

When explaining the game, it may be helpful to model a round using a for loop. For instance, if a player rolled values of **2** and **4**, the for loop would be:

For count = 0, count == **2***4, count+=**2**:
Tap person on the head
Say "duck" or "goose"
Walk by **2** people

Wrap up & reflect

Group students into pairs and have them discuss the following reflection questions. Afterwards, have students share their ideas as a class.

- When do we use for loops in our everyday lives?

  Ex. We can use for loops to model situations repeat instructions a specific amount of times. For instance, we can write sewing two pieces of fabric together with 100 stitches 1/2 inch apart as the following:

  For count = 0, count == 100, count+=1:
  Push needle through pieces of fabric
  Move needle 1/2 inch to the right

- Why are for loops important?

  Ex. Like all loops, for loops help us organize and condense our instructions by only writing the instruction once in a loop. Additionally, for loops are useful for repeating a set of instructions a specific amount of times, and increment through our instructions a particular number of times.