



# Object Attack

## 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 public and private properties to play an action-packed game.

### Prerequisite Knowledge

Student should have completed the Conditional Schedule and Object Oriented Fun activity, which introduces the concept of variables and object-oriented programming.

### Lesson Details

At Decomosphere, students will learn to break problems down into smaller parts with Dot. Students will learn the importance of encapsulation with public and private properties to play an action-packed Grid Attack game.

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

### Learning Objectives

#### Key Question

How can we assign public and private properties to objects?

#### Key Terms

**Encapsulation:** The idea of keeping the variables and functions of an object private

### Curriculum Standards

Students should be able to...

- Explain why encapsulation is used (Decomposition)
- Read, write, and interpret public and private properties (Literacy)
- Use functions and variables to play a Grid Attack game (Creative Arts)

[View standards addressed here](#)



## Lesson Plan

### Materials

- Object Attack worksheet (per student)

### Setup

- Hand out an Object Attack worksheet to each student
- Set up your classroom to form students in groups of 2

## ANSWER KEY & LESSON ANNOTATIONS

Name: \_\_\_\_\_ Date: \_\_\_\_\_

## Object Attack

### Outstanding Objects

Dot is so excited to see you back at Decomosphere! Today, you'll learn more about objects and how they can keep their properties out of sight. Let's jump right in!

### Who let the dogs out?

In object-oriented programming, we can choose to keep the variables and functions of an object private. When these features are private, other objects cannot directly control them. Each object can have a combination of both private and public features.

### What's the difference between private and public?

While private properties (variables and functions) cannot be directly controlled by other objects, public properties can. Let's take a look at an example with Dot's pet dog, Rocky!

Rocky the Labrador			
Variables (Who Rocky is)		Functions (What Rocky does)	
private breed	<b>Why private?</b> Dot can't control Rocky's breed and age.	private bark	<b>Why private?</b> Dot can't naturally control when Rocky barks or sleeps
private age		private sleep	
public owner	<b>Why public?</b> Rocky's owner, Dot, can control who owns Rocky	public feed	<b>Why public?</b> Dot can control when to feed or play with Rocky
		public play	

### Reflect

If we modeled humans as objects, what would some of our private and public properties be?

Ex. Private: Age, name, birthday

Public: Hair color, pets

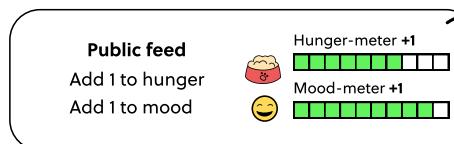
 Decomosphere  
Mission 3

### Why do we use private properties?

In computer science, we use private properties to **protect** features like breed, mood, hunger from being **directly controlled by other objects**.

### How can we change private variables?

Private variables can only be changed through public functions.  
For example, let's take a look at our public function, public feed:



Since our feed function is public (Dot can decide when to feed Rocky), Dot can indirectly control Rocky's private hunger and mood variables.

However, some private variables can never be changed.

For example, we cannot change Rocky's breed as a Labrador.

### Educator Note

Through public functions, we can change private variables like hunger and mood.

Encourage your students to brainstorm other public functions to change the private variable mood of Rocky the Labrador. (Play, walk, pet)



### Decomosphere Mission 3

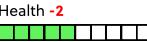
#### Grid Attack!

In this action-packed game, you will be creating your own **player object** with its **private/public variables and functions** to attack an opponent!

#### Materials

- 2 tokens for you and your friend (This will represent where you are on the board!) 
- 1 die 

#### How to Play

- Define your player object's variables by creating a name, age, and health value. Each player starts the game with a health level of 5 (out of 5). Next to each variable and function, circle whether it is private or public! (*Think: Can these properties be directly changed or used by other objects?*)
- Take turns rolling the die with your friend. Use the **walk** function to move your token the number of steps you rolled in any direction(up, down, left, right). Keep track of your activity in your **activity list**.  
*(Ex. If you rolled a 6, you can choose to walk 3 steps left and 3 steps up).*
- If you are on , use the **eat** function to add 1 point to your health.  
  
Health +1
- If you are at the **same location** as your opponent, use the **attack** function to subtract 2 points from your opponent's health.  
  
Health -2
- Winning the game: You win the game when your opponent's health status reaches 0.

#### Extension

If students finish early, have them change the Health variable to a higher/lower number to play again. Print copies of the activity list and game board as needed.

Decomposphere  
Mission 3

My Player Object	
<p><b>Variables</b></p> <p><b>Private/Public Name</b> (circle one) <input type="text"/></p> <p><b>Private/Public Age</b> (circle one) <input type="text"/></p> <p><b>Private/Public Health</b> (circle one) 5 <input type="text"/></p>	<p><b>Functions</b></p> <p> <b>Private/Public Walk</b> (circle one) Move the number of steps rolled</p> <p> <b>Private/Public Attack</b> (circle one) Subtract 2 points from opponent's health</p> <p> <b>Private/Public Eat</b> (circle one) Add 1 point to your own health</p>

My Activity List	
<p><b>Function</b></p> <p>Ex. Walk left 3 steps</p>	<p><b>Health</b></p> <p>5</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
<p><b>Function</b></p>	<p><b>Health</b></p>

Decomposphere  
Mission 3


CS Hands-On
Object Attack
4
CS Hands-On
Object Attack
5



## Wrap up & reflect

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

- Think of an insect. How would you model it as an object with private/public variables and functions?

Ladybug

- Private variables: Number of legs = 6, Number of wings = 2, Color = Red
- Public variables: Location = Park
- Private functions: Fly, Crawl
- Why is it important to establish different properties as either public or private?  
When a property is public, we know that it can be changed directly by other objects. When the property is private, we know that it can either be changed through a public function or simply cannot be changed at all.