

## Lesson 2: Using Methods, Variables, and Parameters

### Try It: Practice Activities

#### Objectives

- Define parameters and how they are used in methods
- Understand the concept of inheritance
- Describe properties of an object
- Describe the purpose of a variable
- Describe programming concepts and define terminology

#### Vocabulary

Identify the vocabulary word for each definition below.

	The memory that the instance of the class has. That memory can be saved and accessed later as long as the instance exists.
	A field used to store information about the class to use immediately or later.
	Used to pass values to methods to specify how objects are to move, or to tell objects what type of response we expect when we ask objects a question.
	Each subclass can use (inherits) the methods from its superclass.
	Word at the beginning of the method that tells us what type of information a method call will return.
	Set of operations or tasks that instances of a class can perform. When a method is invoked, it will perform the operation or task specified in the source code.
	A method call instructs the instance to perform an operation or task. You can read the method to understand what operation or task is to be performed.
	Access, return type, name, and parameters for a method.
	Process of finding and removing bugs—or errors—in a computer program.

## Project

1. Continue the project started in Section 3 Lesson 1.
2. Open the Project named FrogFlyL1. From the Scenario Menu, choose "Save as" and use the name FrogFlyL2. This will keep a copy from Lesson 1 in case there is a need to refer to it later.
3. Compile the Greenfoot project.
4. Add two Frog instances and two Fly instances.
5. Open the code editor for the Frog subclass.
6. In the act() method, write code statements so that the frog moves forward five steps, then turns right 5 degrees.
7. Compile the Greenfoot project, and test the code.
8. Open the code editor for the Fly subclass.
9. In the act() method, write code statements so that the Fly moves forward two steps, then turns left 5 degrees.
10. Compile the Greenfoot project, and test the code.
11. Save the project and exit.

## Optional Practice

1. Open the code editor for an actor subclass. Program instances of the class to move an amount of steps and turn the number of degrees that you specify. Run the scenario to see how the parameters you entered in the Act method impact the instance's movement.
2. In your scenario, add an instance to the world. Inspect the instance's properties. Invoke at least three different methods with void and non-void return types. Invoke two different methods that require parameters.