Java Fundamentals
3-2: Methods, Variables, and Parameters
Practice Activities

Lesson Objectives:

- Define parameters and how they are used in methods
- Understand inheritance
- Describe properties of an object
- Examine the purpose of a variable
- Discuss 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.

Try It/Solve It:

1. State three different methods of an actor. Explain how each of their parameters are used.
2. Open the code editor for an actor subclass. Program instances of the class to move an amount of steps that you specify. Run the scenario to see how the parameters you entered in the act() method impact the instance's movement.
3. Inspect one of your instance's properties. Record its current x and y coordinates.
4. Describe the difference between methods with a void and specific data return type. Add an object to a scenario in Greenfoot, and then invoke at least three different methods with void and non-void return types.
5. Add an object to a scenario, and then invoke two different methods that require parameters. Write down the method signature, and what happened to the instance after the method was invoked.
6. In your scenario, program an instance of a class to move 3 steps, and then turn 18 degrees. Compile the code, and then test the results of your programming statements.
7. True or false: empty parameter lists will need data to invoke the method.