Lesson 3.03 — Return Values

Overview

Objectives — *Students will be able to...*

- **Write** a program that returns values.

Assessments — *Students will...*

- **Complete** Practice questions
- **Write** a program to meet a Pokémon Challenge

Homework — *Students will...*

- **Complete** chapter 3 self-check question 17 and exercise #1

Materials & Prep

- **Projector and computer** (if you are able to/opt to use Eclipse with your students)
- **Whiteboard and markers**
- **Poster or image** of
  - Blastoise (http://tinyurl.com/ndy3v69) and
  - Raichu (http://tinyurl.com/n2u5vn2)

This lesson uses Pokémon code; you should read through the example and learn how stats work at (http://bulbapedia.bulbagarden.net/wiki/Stats). The wiki offers a great amount of detail, and can be used to offer additional programming challenges to advanced students. You may want to bookmark this page for future reference, since Pokémon stats are used in future examples.

Pacing Guide

Section	Total Time
Bell-work and attendance	5min
Review and intro to returning values	15min
Practice	15min
Pokémon Challenge	15min

Procedure

This lesson introduces methods that return values, and familiarizes students with the Math class. You should hook students by introducing the Pokemon challenge (to be completed at the end of the class); students will create more code to be used in their larger Pokemon program.

Bell-work and Attendance [5 minutes]

Review and Introduction to Returning Values [15 minutes]

1. Begin with a lecture/discussion about the Pokémon challenge and returning values.
   - When you're playing a video game like Pokemon, part of the fun is the graphic images that help communicate the story of your battles, training, or travels. However, all of the outcomes of your game are determined by math that happens in the background.
   - The Pokemon with better stats wins a battle; some additional random numbers are thrown in to represent the unpredictable nature of the real world.
   - Today we're going to learn how to write methods that **return a value**—we already know how to get Java to compute simple equations for us; now we're going to learn how to get Java to give us back those numbers so we can use them elsewhere in our program.

- While programmers can manipulate the parameters passed into a function, their code is operated on a copy of the value, and not the value itself.
- If int x is passed as an argument into an expression as the parameter int num, the function may manipulate the value stored in num. When the function returns, x will be unchanged.
- Students may find the following analogy from StackOverflow helpful: "The procedure defines a parameter, and the calling code passes an argument to that parameter. You can think of the **p**arameter as a **p**arking space, and the **a**rgument as an **a**utomobile."

- We're also going to learn how to use a collection of equations that Java already has written for us called the Math Class. These pre-made methods make doing complex equations much easier!

2. Here's the syntax to writing a method that returns the sum of numbers 1 – n; first the header:

   ```
   public staticdoublesum (double n) {
   ```

   - We're used to having "void" in this spot—but void actually means that we're writing a method that we don't expect to return anything. In this highlighted example, we write our method to return a value of the double type.

     ```
     public static doublesum(double n) {
     ```

   - Highlighted is our method name, it goes in the same place in the header as it always did.

     ```
     public static double sum (double n) {
     ```

   - We used to just leave the highlighted section as empty parentheses, now we have to tell Java what type of type of data we're going to put into the method (the formal parameter).

3. Ask students to change the method header so it sums data from input int data and returns data of the type int as well.
4. Ask students to change the method header so this new method is called doubleSum.
5. If students are adjusting these parts of the method header with ease, move onto the method body:

   ```
   public static int sum (int n) {
       return (n + 1) * n / 2;
   ```

   - Without the special return statement, this wouldn't return a value to the main method! It would basically be a void method, like the ones we wrote before. It is an error in Java for flow of control to reach the end of a non-void method without a return!
   - This method only makes sense if we have a main method that can pick up the value that we're asking Java to return, so have students write a main method:

     ```
     public static void main (string[] args) {    // a. Why is the main method void?
         int answer = sum (100);                  // b. What is this line doing?
         System.out.println ("The sum of 1 to 100 is" + " " + answer);
     ```

     1. The method main is void because it returns no value.
     2. This line is declaring & assigning a value to the variable `answer`.

6. Ask students to tell you where to place the brackets, and briefly review scope.

If they want to do fancier math, they can use the formulas that Java has already stored in the Math class. As students read last night, there is a list of the most-used formulas in table 3.2 of the book.

- There is special notation needed for the methods in the Math class, because you have to tell Java to go and use the method in another class. We call this "dot notation."
- The Math class is an Application Program Interface (API). APIs are libraries that provide fuctionality to simplify complex programming tasks.
- Documentation for APIs and libraries are essential to understanding the attributes and behaviors of an object of a class. Java's Javadoc describes many of the Java APIs. For example, the Math API: https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html

- If you wanted to generate a random number to use in a formula for your Pokemon game (to add a little chance to a battle, lets say), you would create a method:

```
public static double pokemonRandom() {
    return Math.random() * 100;
}
```

- The math class' method random gives a random number between 0.0 and 1.0; we multiply it by 100 because Pokemon random numbers are values between 0 and 100. This method now gives us a random number between 0 and 100. We can use our new pokemonRandom method whenever we need a random number from that range.

If students are getting the material, have them work independently on the practice problems, otherwise, work through the problems together as a whole class.

Practice- [15 minutes]

Have students work individually or in pairs to complete the following practice self check questions:

1. Self-Check 3.6: parameterMysteryNumbers
2. Self-Check 3.13: mathExpressions
3. Self-Check 3.16: min

Pokémon Challenge [15 minutes]

Once students have completed these exercises, invite them to complete the following Pokémon challenge:

POKEMON CHALLENGE:

A Pokémon's base stat values will most often have the greatest influence over their specific stats at any level. If we leave out individual values, effort values, and nature, a level 100 Pokémon's stats in Attack, Defense, Speed, Special Attack, and Special Defense will be exactly 5 more than double its base stat values in each, while the Hit Points (HP) stat will be 110 plus double the base stat value (except in the case of Shedinja, whose HP is always 1).

Write a program that returns a Pokémon's stats for Attack and HP at level 100. You should use parameters and methods that return values for this program. You may choose to use the base stats for Blastoise OR Raichu given here:

Blastoise	Raichu
HP: 79	*HP:* 60
Attack: 83	*Attack:* 90
Defense: 100	*Defense:* 55
Special Attack: 85	*Special Attack:* 90
Special Defense: 105	*Special Defense:* 80
Speed: 78	*Speed:* 110

Course Final Project Pay close attention to the base stats for the Pokemons and how health values are deducted based on the attacks.

When you complete your final project in the course, this information will be very important as you apply these concepts to a new battle context that will be a little different than Pokemon.

Accommodation and Differentiation

For students who complete the Pokemon challenge early, ask them to flesh out their program by:

- Adding methods that return stats for Speed, Special Attack, Special Defense, and Defense.
- Writing a method that will compare stats between Blastiose and Raichu, then return the maximum value. (This program doesn't need to accept user input –yet!)

If students are struggling with the Pokemon Challenge, urge them to begin with their structure diagram of pseudocode. Once they have this code, help them write the method to calculate stats by assisting with the algebra, if needed.

Misconceptions

- Output to the console is somehow synonymous with the return value of a method: overloading the use of the word output.

Video

- BJP 3-2, *Parameters and Return Values* http://media.pearsoncmg.com/aw/aw_reges_bjp_2/videoPlayer.php?id=c3-2
- CSE 142, *Methods that Return Values* (1:25-8:45) https://www.youtube.com/watch?v=kCqcmWk8qoY&start=85
- CSE 142: *Return value worked example* (8:46-38:32) https://www.youtube.com/watch?v=kCqcmWk8qoY&start=526

Forum discussion

Lesson 3.03 Return Values (TEALS Discourse account required)