

## Activity 4: Methods and Stack Diagrams

### Team Roles (5 min)

Decide who will be what role for today's activity. List the names of your team members next to their role.

Manager:
Spokesperson:
Quality Control/Recorder:
Process Analyst:

If you have only three people, one may serve as both manager and spokesperson. If you have five team members, then assign two people to process analyst.

1. For each role, give an example of how someone observing your group would know that a person is not doing their job well.

- Manager
- Spokesperson
- Quality Control
- Process Analyst

## Model 1: Call and Return (15 min)

Each statement in this program is a *method call*. At the end of a method, the program *returns* to where that method was called.

```
public class Model1 {  
    public static void main(String[] args) {  
        System.out.println("First line.");  
        threeLine();  
        System.out.println("Second line.");  
    }  
  
    public static void newLine() {  
        System.out.println();  
    }  
  
    public static void threeLine() {  
        newLine();  
        newLine();  
        newLine();  
    }  
}
```

```
CALL println  
RETURN to main  
CALL threeLine  
    CALL newLine  
        CALL println  
            RETURN to newLine  
        RETURN to threeLine  
    CALL newLine  
        CALL println  
            RETURN to newLine  
        RETURN to threeLine  
    CALL newLine  
        CALL println  
            RETURN to newLine  
        RETURN to threeLine  
    RETURN to main  
CALL println  
RETURN to main
```

## Questions

2. How many lines of code call the `System.out.println` method?
3. How many times is `println` actually called when the program runs?
4. For each `CALL` in the program trace on the right, draw an arrow to the corresponding method call on the left.
5. What is the output of the program? Please write `\n` to show each newline.

6. In your own words, describe what methods are for. Why not just write everything in main?

7. What is the difference between a method and a variable? What do they have in common?

8. When Java sees a name like `x`, `count`, or `newLine`, how can it tell whether it's a variable or a method? (Hint: syntax)

## Model 2: Order of Execution (10 min)

```
public class Model2 {  
    public static void baffle() {  
        System.out.print("wug");  
        ping();  
    }  
  
    public static void main(String[] args) {  
        System.out.print("No, I ");  
        zoop();  
        System.out.print("I ");  
        baffle();  
    }  
  
    public static void ping() {  
        System.out.println(".");  
    }  
  
    public static void zoop() {  
        baffle();  
        System.out.print("You wugga ");  
        baffle();  
    }  
}
```

### Questions

9. How many method call statements are in this program?
10. In what order are the methods declared in the class?
11. Based on Model 1, describe the order in which methods actually run. What does that order have to do with your answer to #9?
12. What is the output of the program? Be precise about where there are spaces and where there are newlines.

### Model 3: Parameters and Arguments (10 min)

When declaring a method, you must specify its *parameters* in parentheses. A parameter is a variable required by the method. When calling a method, you must provide the value for each parameter. We refer to the actual values as *arguments*.

Method Declarations	Method Calls
<code>public void println(String x)</code>	<code>System.out.println("Price: " + price);</code>
<code>public static void repeat(String word, int count)</code>	<code>repeat("Hello " + name, 7);</code>
<code>public static void show(String line)</code>	<code>show("The " + price + " is " + price);</code>

### Questions

13. How many parameters and arguments does each method have?

Method	# Params	# Args
<code>println</code>		
<code>repeat</code>		
<code>show</code>		

14. Assuming `price` is the double `9.99`, what is the argument for `println`?

15. Implement the `showLength` method below based on the following examples:

Method Call	Output
<code>showLength("CS149");</code>	CS149 is 5 characters long
<code>showLength("C:\\Users\\Colbie");</code>	C:\Users\Colbie is 15 characters long

Recall that Strings have a `length` method that takes no arguments and returns the length of the string.

```
public static void showLength(String s) {  
  
}
```

16. Write a method `printTime` that takes two integers, `hours` and `minutes`, and displays them separated by a colon. For example, `printTime(11, 59)` should print the text `11:59` followed by a newline.