

## Lesson 1.06 — Static Methods & Method Calls (2/2)

### Overview

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

- Use structure diagrams to plan complex programs.
- Manage complexity by using method calls.

Assessments — *Students will...*

- Write a structured Pikachu program

Homework — *Students will...*

- Outline Ch.1
- Complete Programming Project #1\* & #3 (must include a structure diagram for each)

### Materials & Prep

- Projector and computer (if you are able to/opt to use Eclipse with your students)
- Whiteboard and markers
- Rosters for class teams
- Ball (preferably a large inflatable ball or kickball)

The teams for today's competition should be your best guess at tiered grouping (these groups will probably change as the year goes on and you learn more about your students). Depending on the size of your class, you should aim for 4 teams or teams of 4 people.

### Pacing Guide

Section	Total Time
Bell-work and attendance	5min
Introduction & challenge	10min
Review/re-teach	5–15min
Pikachu challenge	10–15min
Students trade work, check, and turn in	5min

### Procedure

Today's class re-teaches the many concepts introduced during yesterday's lesson on decomposition, static methods, and methods that call other methods and goes over control flow. Since different students will progress at different rates, we'll begin this lesson with an assessment (in the form of a competition) to determine how much re-teaching you need to do. The competition is really group work in disguise, and will encourage students to teach and help each other while writing the sample program.

**Bell-work and Attendance [5 minutes]**

**Introduction and Challenge [10 minutes]**

1. As your hook, grandly announce a class competition between teams and announce the prize for the winning team (this might be TEALS swag, bonus classroom participation points, or additional raffle entries in the year-end TEALS giveaway).
2. Announce class teams and rearrange students as needed.
3. Announce the 10-minute time limit, set the timer, then reveal the challenge on printed handouts or on the projector with great fanfare. **CHALLENGE:** Write a Java program called StarFigures that generates the

following output. You MUST include a structure diagram or your answer will be disqualified. A correct answer will use static methods to show structure and eliminate redundancy in your solution.

```
*****
*****
 * *
 *
 * *

*****
*****
 * *
 *
 * *
*****
*****

 *
 *
 *
*****
*****
 * *
 *
 * *
```

4. Give students 10 minutes to complete the challenge, and take note of which team finishes first. If students are struggling, you may extend the time, or offer universal helpful tips. The team that has the **correct answer first** wins the prize.

### Review/Teach [5-15 minutes]

Review student answers together as a whole group, revisiting concepts taught earlier in the week as mistakes come up. Whenever possible, have students volunteer the correct procedure, approach, or code. Encourage students to take notes during this process so they can review topics over the weekend.

### Review/Teach OPTIONAL INSTRUCTION

If students are having trouble understanding the flow of control, you can do this physical activity with them (bonus: this activity can be built on later in the year when discussing return values).

1. Write or project the following code as an example (have students help you with the headers if you are writing):

```
public class SquarePants {

    public static void spongebob() {
        System.out.println("Well, it might be stupid");
    }

    public static void patrick() {
        System.out.print("but it's also");
    }

    public static void squidward() {
        System.out.println("dumb.");
    }

    public static void main (String[] args){
        spongebob();
    }
}
```

```

        patrick();
        squidward();
    }
}

```

2. Assign a student to each of the methods: **main**, **spongebob**, **patrick**, and **squidward**, and have them come to the front of the classroom. Have another student act as Java; tell them they are to write the output on the board (or type it if you're using a computer/projector system). From here on out you should address students by their component names (so always call student acting as method **patrick** as Patrick, and so on).
3. Narrate the flow of control as you toss the ball to **main**. Have **main** pass the ball to **spongebob** (and make sure Java "outputs" Well, it might be stupid on the Whiteboard before control passes back to **Main**).
  - Discuss with the class how Java knows to return to the next call in **main** (the close-bracket), and have the class direct control to **patrick**, then **main**, then **squidward** as Java writes the output on the board.
  - Pay special attention to the print statement in the **squidward** method, and if kids miss it, make some sort of "error" noise. Bonus: encourage the students to make the error noise for you.

### Pikachu Challenge [10-15 minutes]

On the board or projector, have the students finish the class with the Pikachu challenge:

*Write a program called `PikachuChatter` that outputs the following:*

```

Pika pika pika chu pika chu peeeee ka pika chu!
(\__/)
(o^.^)
z_(")(")

```

```

Pika? Pika pika pika chu peeeee ka chu!
(\__/)
(o^.^)
z_(")(")

```

- You can reuse the code that you wrote earlier this week.
- Your program should use static methods and method calls.
- You should include a comment at the start of the program that explains what the code does (you might want to use this code later in the year when we build a larger Pokémon program).

### Students trade work, check, and turn in [5 minutes]

At the end of class, have students review one another's Pikachu challenge codes before submitting.

### Accommodation and Differentiation

If you have students who are speeding through this lesson, you should encourage them to:

- Complete the remaining problems:
  1. Self-Check 1.24: Strange2
  2. Self-Check 1.25: Strange3
  3. Self-Check 1.27: Confusing2
  4. Self-Check 1.28: Confusing3 questions.
- Have the student write a sample test question with output that can be written using method calls. Be sure they include the answer key with the sample question!

If you have the good fortune of not needing to re-teach any concepts, you can magnanimously give students extra time to start on the homework programming project #1. If you are doing a lot of re-teaching during this class, and you feel that students need the emotional reward, you may drop programming project #1.

## Video

- BJP 1–4, *Drawing Complex Figures with Static Methods* [http://media.pearsoncmg.com/aw/aw\\_reges\\_bjp\\_2/videoPlayer.php?id=c1-4](http://media.pearsoncmg.com/aw/aw_reges_bjp_2/videoPlayer.php?id=c1-4)

## Forum discussion

Lesson 1.06 Static Methods and Method Calls (TEALS Discourse account required)