Lesson 3.02 — Limitations of Parameters & Multiple Parameters

Overview

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

- **Modify** programs using parameters and class constants to create original artworks.

Assessments — *Students will...*

- **Complete** an art project and "artist statement" justifying their programming choices

Homework — *Students will...*

- **Read** BJP 3.1 "Parameters versus Constants" and "Overloading Methods"
- **Jazz up** art projects and programs

Materials & Prep

- **Projector and computer**
- **Whiteboard and markers**
- **Sample** of final project (picture, artist statement, code)
- **Pictures** from previous year's "gallery opening"
- **Ball** for passing flow of control

Around this time of year, many schools have open houses and/or parent-teacher conferences. If you have the time and resources, this programming project is a excellent opportunity for students to showcase their work for parents, principals, superintendents, etc. For instructions on using this lesson as an opportunity to share student work, refer to Accommodation & Differentiation at the end of this lesson plan.

If this is your first year teaching this lesson, you obviously won't have pictures from previous years events. Make sure to document this lesson for future years so you can hook students with the glamour and excitement of an art gallery opening.

Pacing Guide

Section	Total Time
Bell-work and attendance	5min
Review and intro to multiple parameters	15min
Student practice	30min

Procedure

Hook your students by showing an example of a completed art project, statement, and code sample. If you don't have previous work to share with students, show them some samples of ASCII art and ask students to predict how many lines of code they'd need to design the images.

Bell-work and Attendance [5 minutes]

Review and Introduction to Multiple Parameters [15 minutes]

1. If you will be holding a reception for student art (see end of lesson for details), announce that now. Segue into today's lesson by explaining that you're going to give students a few additional tools to use in their art project. Give them a quick recap/review before teaching **multiple parameters**:

   - Call students up to the board—some to help you demonstrate flow of control, another to produce console output on the whiteboard.

   - Using the example below, review the flow of control by having the students (main and parameter) pass information to you (the method). Have the class direct your other student what output to write on the

board. Ask students what variable x is throughout the example to drive home the idea that parameters don't change value when the method changes a local value.

```java
public class ParameterExample {
    public static void main (String[] args) {
        int x = 17;
        doubleNumber(x);
        System.out.println("x = " + x);
        System.out.println();
        int number = 42;
        doubleNumber(number);
        System.out.println("number = " + number);
    }
    public static void doubleNumber (int number) {
        System.out.println("Initial value = " + number);
        number = number * 2;
        System.out.println("Final value = " + number);
    }
}
```

- It is possible to declare multiple parameters! The trick is to always make sure your method accepts the parameters in the same order. When calling the method, pass the parameters in the same order in which they were declared.

```java
public static <type> <name> (<type> <name>, <type> <name> … <type> <name>) {
    <statement>;
    <statement>;
    …
    <statement>;
}
```

2. Using the diagram below, walk students through this coding example. You should have students draw, circle, and explain the parts of the code in their own notebooks so they have a chance to think this through.

```java
public static void main (String [] args) {

    writeChars ('=', 20);   //----------------- Writes a line of 20 ='s
    System.out.println();   //----------------- Returns to the next line

    for (int i = 1; i <= 10; i++) {  //------- For 10 lines of picture (height)
        writeChars('>', i);          //------- Increase the number of '>'s in each line
        writeChars(' ', 20 - 2*i);   //------- Decrease the number of spaces in each line
        writeChars('<', i);          //------- Increase the number of '<'s in each line

        System.out.println();        //------- Go to the next line before starting the body
    }                                //         of the loop again.
}

public static void writeChars (char ch, int number) {
    for (int i = 1; i <= number; i++) {
        System.out.print(ch);
    }
}
```

- Have students predict how the output will change if you change 1 or 2 things in the code, then allow them to start their open-ended activity.

Student Practice [30 minutes]

1. Encourage students to spend a few minutes fiddling with the code we reviewed in class to see how it changes the outputted image. It's really important that students have time and space to fiddle with code to understand how the different parts relate. Encourage students to start with this code and build from it/change it to make whatever images they want to make for their final project.

2. Instruct students that they will need to create a design that uses parameters, loops, and/or nested loops. Extra points will be awarded for particularly complex/creative images.

3. They should use (and keep copies of) a structure diagram and/or pseudocode for their own records, since they will be expected to write a short paragraph explaining in plain English (or their native tongue).

4. They are allowed to continue working on this project outside of class: you have graciously decided not to assign problems for homework tonight so that they will have time to work on the code.

If you find that students are highly engaged in the project, you might opt to extend this lesson to a 2 period coding session. The more opportunity students have to manipulate code and check output, the more intuition they will have down the road as they internalize what individual pieces of code mean.

Accommodation and Differentiation

If you feel that students need the extra review, ask students to help you build the example program you use during the introduction of class. Do this as a whole group to keep class on pace, and write only the information that students give you (i.e. let mistakes happen, then guide the class to error-check their own work).

If students are having a hard time with the coding project, you may encourage them to work in pairs instead of as individuals. If some of your class partners up, make sure that the class understands that teams will be expected to write more complex images than those created by students working alone.

Encourage students to think about their learning and coding processes by having students articulate HOW they've learned what they learned. This can be done by having students explain their coding decisions to peers and laypeople with no coding experience. A great way to do this in a low-pressure setting is to have students share their work with parents and others at an open house; ham it up a little for fun!

- Display student work around the room, and have students explain or talk about their work during an "art gallery opening" held at lunch, after school, or in the evening during parent teacher conferences or open house events.

- Invite your principal, superintendent, and school staff to attend—this is a great way to increase community buy-in for your class.

- Provide (or have students bring in) cubed cheese, fruit, crackers, white grape juice, and crackers. Create a fancy reception table with a table cloth, plastic wine glasses, and cocktail napkins.

- Play jazz, string quartet, or trip hop softly in the background to create a trendy reception vibe, and have students "dress like artists" and stand near their work. Make sure you dress up/down to fit the scene!

- If possible, hand visitors "question cards" to give them ideas of how they can learn more about student work as they circle the "gallery." Cards should include prompts like:
  - Tell me about your picture; how did you come up with this idea?
  - Can you explain to me a bit about how you get the computer to draw this image?
  - How did you go about the process of writing this code?
  - How did you start this project?
  - If I wanted to change this picture, how would I go about doing it?

An event like this maintains/raises student morale as students: - take ownership of their work, - receive praise for their hard-earned accomplishments thus far, and - realize how much they've learned when they explain code to a non-coder.

Forum discussion

Lesson 3.02 Limitations of Parameters & Multiple Parameters (TEALS Discourse account required)