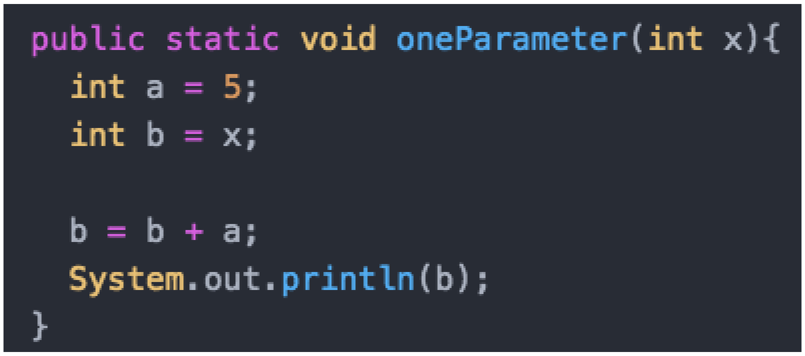
Activity Title: **Subject**: Computer Science **Grade**: 8th	**Teacher**: Ms Wingreen **Duration**: 45 mins

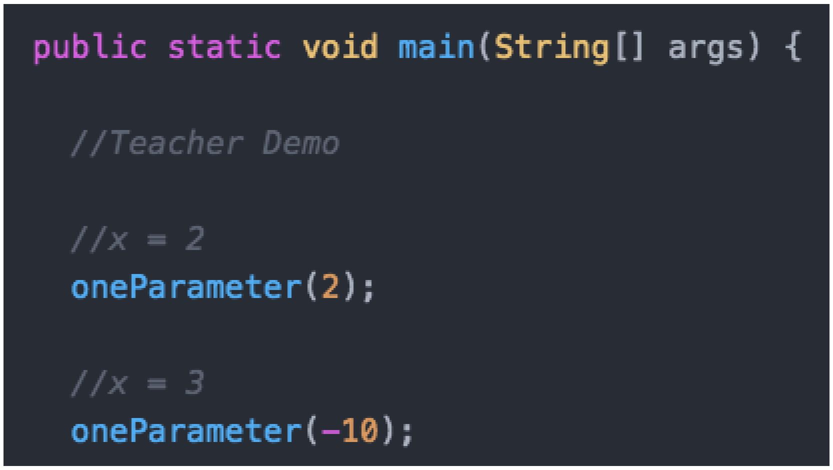
Summary/Timeline of Activity:

- **Do Now**:
    - Students solve each other's trace tables from the previous day
        - In pairs, students will present their birds-eye view pictures of the unplugged activity from the previous day's lesson. They will explain the methods that they selected and walk their partner through their trace tables.
        - Students will quiz each other by giving their partner an input and a method call and having them trace out the result.

- **Mini-lesson**:
    - Teacher demos what it looks like in the code to call a method with one parameter.
    - Trace through calling oneParameter() with x = 2 and x = -10. Students should assist in figuring out what happens on each line:
        - What is the value of a?
        - What is the value of b?
        - What is the new value of b?
        - What does the function print out?

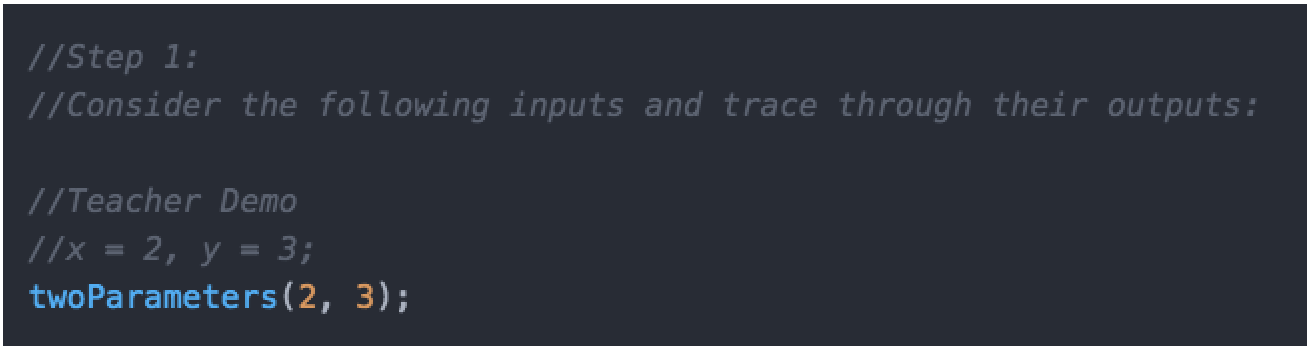


```
public static void oneParameter(int x){
  int a = 5;
  int b = x;

  b = b + a;
  System.out.println(b);
}
```

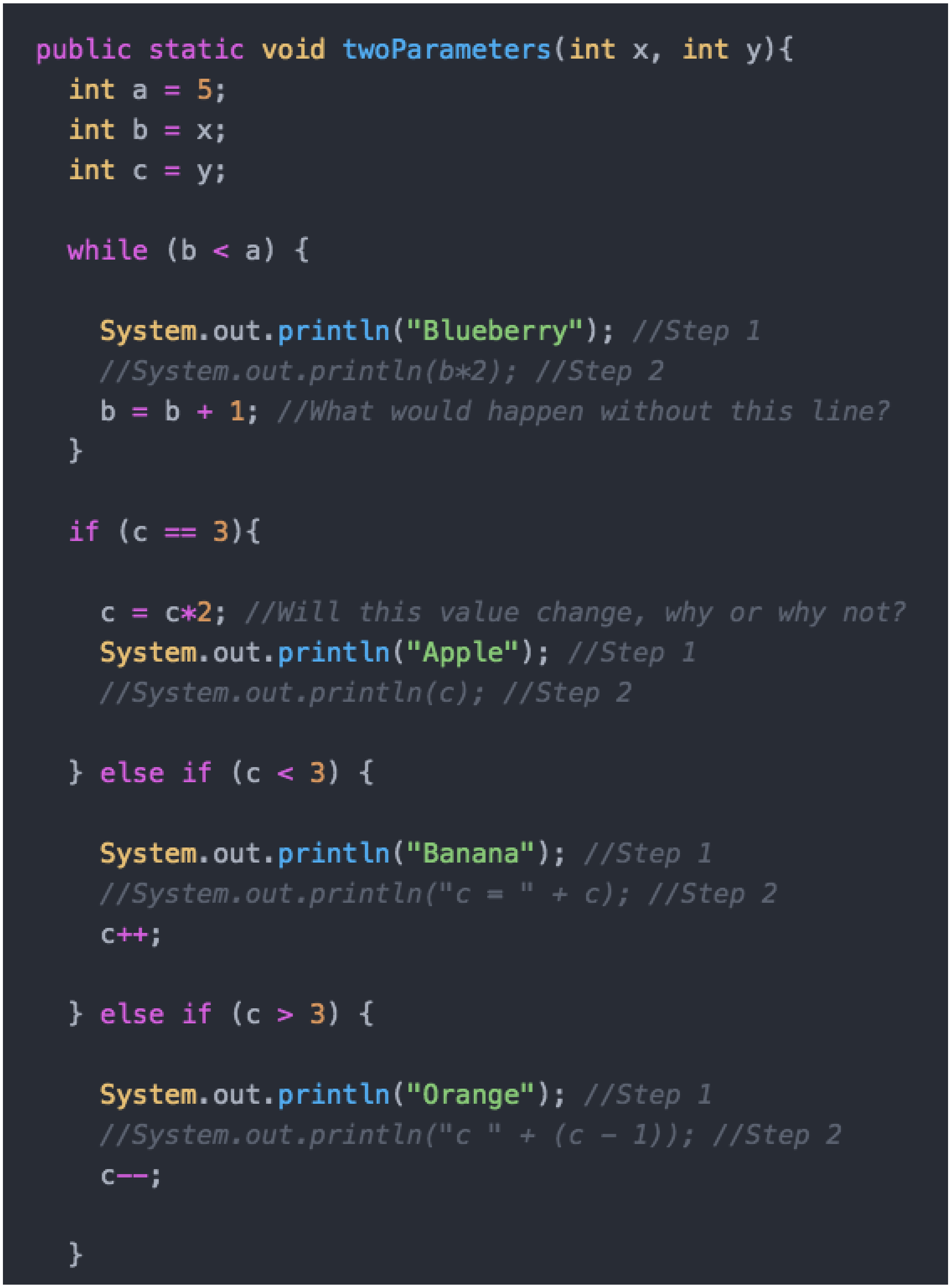


```
public static void main(String[] args) {

  //Teacher Demo

  //x = 2
  oneParameter(2);

  //x = 3
  oneParameter(-10);
```

- **Mini-lesson (cont.)**:
    - Teacher demos what it looks like in the code to call a method with two parameters.
    - Trace through calling oneParameter() with x = 2 and y = 3. Students should assist in figuring out what happens on each line:
        - Ask students “What would happen without this line?” on line 24
        - Ask students “Will this value ever change, why or why not?” on line 29
    - Then ask after tracing through:
        - What is the value of x?
        - What is the value of y?
        - What is the value of a?
        - What is the value of b?
        - What is the value of c?
        - Which if any of the variables will remain constant?
    - As teacher is tracing through the code, students should be completing their trace tables (worksheets below)



```
//Step 1:
//Consider the following inputs and trace through their outputs:

//Teacher Demo
//x = 2, y = 3;
twoParameters(2, 3);
```



```
public static void twoParameters(int x, int y){
  int a = 5;
  int b = x;
  int c = y;

  while (b < a) {

    System.out.println("Blueberry"); //Step 1
    //System.out.println(b*2); //Step 2
    b = b + 1; //What would happen without this line?
  }

  if (c == 3){

    c = c*2; //Will this value change, why or why not?
    System.out.println("Apple"); //Step 1
    //System.out.println(c); //Step 2

  } else if (c < 3) {

    System.out.println("Banana"); //Step 1
    //System.out.println("c = " + c); //Step 2
    c++;

  } else if (c > 3) {

    System.out.println("Orange"); //Step 1
    //System.out.println("c " + (c - 1)); //Step 2
    c--;

  }
```

- **Independent Work Time**:
    - Students work in pairs to trace through the remaining examples in the code file
    - Extension: Modify the methods and trace through using updated values

- **Exit Ticket**:
    - Completed trace tables

<table>
<tr><th colspan="3">Desired Results</th></tr>
<tr>
<td><b>Common Core Standards:</b>:
● Design or remix a program that uses a variable to maintain the current value of a key piece of information.
● The focus is on understanding that variables can be used to track the value of a concept in a program as it changes over time.</td>
<td><b>Vocabulary</b>:
● Method
● Parameter
● Value
● Store
● Trace</td>
<td><b>Resources</b> (provide URLs):
● LessonPlan2_code.java
● Blank trace tables (used for Independent Work / Exit Ticket)</td>
</tr>
<tr>
<td colspan="2"><b>Learning Objectives:</b>
The student will be able to…
● Understand the difference between methods that take one or two parameters
● Trace (in code) how the variable’s value changes when different methods are applied</td>
<td><b>End product</b>:
The student will create…
● A solution to another student’s work from the previous day
● A trace table for each method call in the code file</td>
</tr>
</table>

```
//x = 2
oneParameter(2);
```

Variable x		
Input	Method	Output

Variable a		
Input	Method	Output

Variable b		
Input	Method	Output

```
//x = -10
oneParameter(-10);
```

Variable x		
Input	Method	Output

Variable a		
Input	Method	Output

Variable b		
Input	Method	Output

```
//x = 2, y = 3;
twoParameters(2, 3);
```

Variable x			
	Input	**Method**	**Output**
Step 1:			
Step 2:			

Variable y			
	Input	**Method**	**Output**
Step 1:			
Step 2:			

Variable a			
	Input	**Method**	**Output**
Step 1:			
Step 2:			

Variable b			
	Input	**Method**	**Output**
Step 1:			
Step 2:			

Variable c			
	Input	**Method**	**Output**
Step 1:			
Step 2:			

Which if any of the variables will remain constant?

```
//int x = 4, y = 1;
twoParameters(4, 1);
```

Variable x			
	Input	Method	Output
Step 1:			
Step 2:			

Variable y			
	Input	Method	Output
Step 1:			
Step 2:			

Variable a			
	Input	Method	Output
Step 1:			
Step 2:			

Variable b			
	Input	Method	Output
Step 1:			
Step 2:			

Variable c			
	Input	Method	Output
Step 1:			
Step 2:			

Which if any of the variables will remain constant?

```
//int x = 5, y = 5;
twoParameters(5, 5);
```

Variable x

	Input	**Method**	**Output**
Step 1:			
Step 2:			

Variable y

	Input	**Method**	**Output**
Step 1:			
Step 2:			

Variable a

	Input	**Method**	**Output**
Step 1:			
Step 2:			

Variable b

	Input	**Method**	**Output**
Step 1:			
Step 2:			

Variable c

	Input	**Method**	**Output**
Step 1:			
Step 2:			

Which if any of the variables will remain constant?