# WaterAndPitchersObjects.docx

## Introduction

Water and pitchers is a classic logic game. You have a 5-oz. pitcher, an 8-oz. pitcher, and a spigot that supplies an endless amount of water. There are no markings on either pitcher. You are limited to these operations:

- Fill a pitcher to capacity.
- Transfer the contents from one pitcher to the other (no spilling allowed).
- Empty the contents of a pitcher.

Assuming both pitchers are empty at the start, can you use any of the listed operations to get 3 oz. of water in either pitcher? Continue the above operations to get these amounts (order is unimportant): 1, 2, 4, 6, 7. First, **solve this puzzle using paper and pencil**.

## Computer Modeling

Obviously, this puzzle does not require a computer to solve, but It does provide an opportunity to model the problem and program it.

Let's represent the first pitcher with the integer variable X5 and the second pitcher with the integer variable Y. We have these invariants within the program.

- $0 \le X5 \le 5$
- $0 \le Y8 \le 8$

This suggests two global variables:

```
static int X5 = 0; // five-ounce pitcher
static int Y8 = 0; // eight-ounce pitcher
```

That's all of the data required for the program. Let's model the operations by some simple methods. We can fill the first pitcher with water. This can be modeled by method `fillX5()` as follow:

```
static void fillX5() {
	X5 = 5;
	displayX5andY8();
}
```

Notice that at the end, this method calls displayX5andY8() method which simply prints the current X5 and Y8 values:

```
static void displayX5andY8() {
	System.out.printf("(%d,%d)\n", X5, Y8);
}
```

If we do this following every operation, we will be able to see the exact values (ounces) in each pitcher.

Here are the method signatures for the rest of the operations (you must provide the code):

```
static void fillY8()
static void emptyX5()
static void emptyY8()
```

```
static void transferX5toY8()
static void transferY8toX5()
```

The only tricky methods are the transfer operations. To get it right, you have to calculate the remaining capacity of the destination pitcher when the method is called. That's the amount that can be transferred due to the "no spilling" rule. Hint: use the `Math.min()` method.

The `main()` method should be a series of calls to any of these methods with the goal of producing one of the desired amounts. The easiest goal is the 3-oz. goal which can be produced as follows:

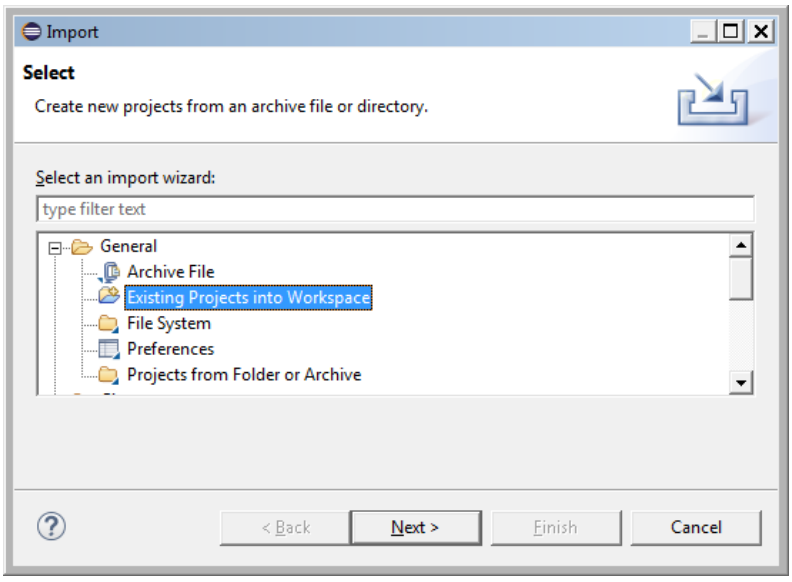
```
public static void main(String[] args) {
	fillY8();
	transferY8toX5();
	System.out.println("---*-");  // (5,3)
// ...more operations to produce the other amounts
```

Notice the print statement that highlights the first solution. Your program should do this for the benefit of the end-user of your program. Here are the remaining solutions with the intermediate steps **not** shown (your steps may differ):

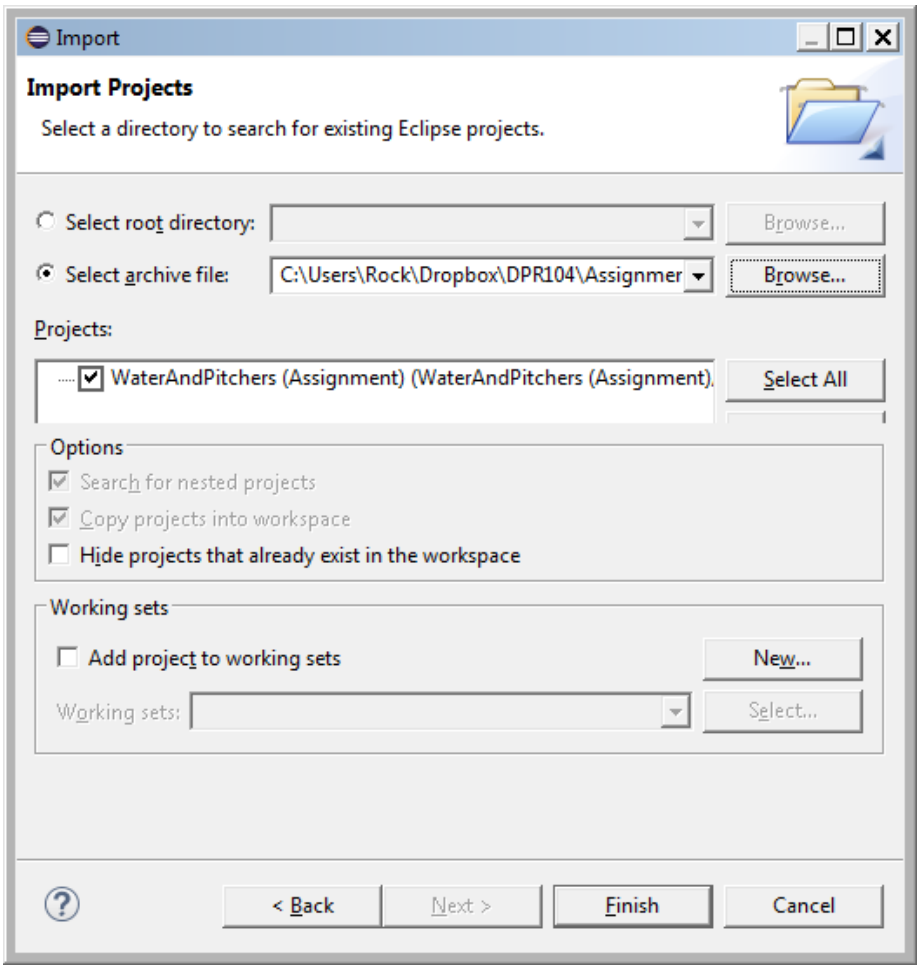
```
(5,3)
---*-
(2,8)
-*--
(5,6)
---*-
(5,1)
---*-
(5,4)
--_*-
(5,7)
---*-
```

## Assignment

Download the starter project (WaterAndPitchers.zip) from Canvas and import into Eclipse as follows. From the file menu select *File => Import...* . In the dialog select *Existing Projects into Workspace* then click *Next*.




On the next screen click the *Select archive file* radio button then use the *Browse...* button to navigate to where you downloaded the zip file. Select the file and click *Open*. At this point the dialog should look like this:



Click *Finish*.

When you have completed the assignment attach the Main.java file to your Canvas submission. Make sure that your name appears as a comment at the top of the file.

```
// Robert Rodini
public class Main {
```

Hopefully you have found all of the solutions to this classic puzzle.