In this exercise you'll be practicing more loops and also getting some experience with using a graphical "grid" object that you can paint. We will use a similar grid in the next project.

1. You've checked out Lab-Grid to get this PDF.

2. Open the file called **OperatorMaker.java** and implement the four empty "helper" methods that you see in this file as well as the **drawOp** method. Each of the methods accepts a parameter that is of type **SquareGrid**. This parameter has already been initialized before your method was called, so do not attempt to create your own **SquareGrid** object**.** When you are running the **ExampleDriver** in Eclipse the grid will be created there and passed into the **drawOp** method, and the **drawOp** method will then pass it along to the appropriate helper method. **The size of the grid may vary, but it will always be an odd number.**

   A **SquareGrid** is like the grid class that you will see in Projects that ask you to draw shapes in a Grid. Each method you write will draw a different symbol on the grid. **See the back page for pictures of the symbols that you will be drawing on the grid.**

   To obtain the size of the grid, use the following syntax:

   ```
   int size = grid.getHt();     // Sets the variable "size" equal to the size of the grid.
   ```

   To draw on the grid, use the following syntax:

   ```
   grid.setColor(row, column, Color.BLUE);    // That will paint the cell blue.
   ```

   The "brush stroke" will always be of width 1 in this lab.

HINT: Rather than writing redundant code, you can have some of your methods make calls to others. For example, your "plus" method could start by calling "minus" and then just drawing a vertical line!

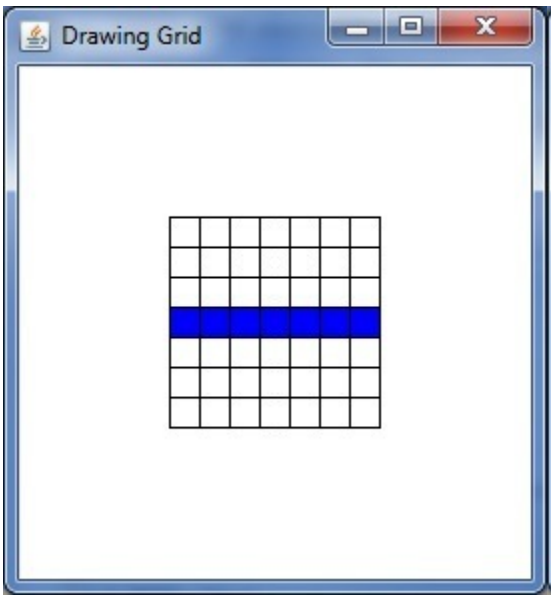
3. To test your code, run the main method that is in the class called **ExampleDriver**.

4. Submit your work!

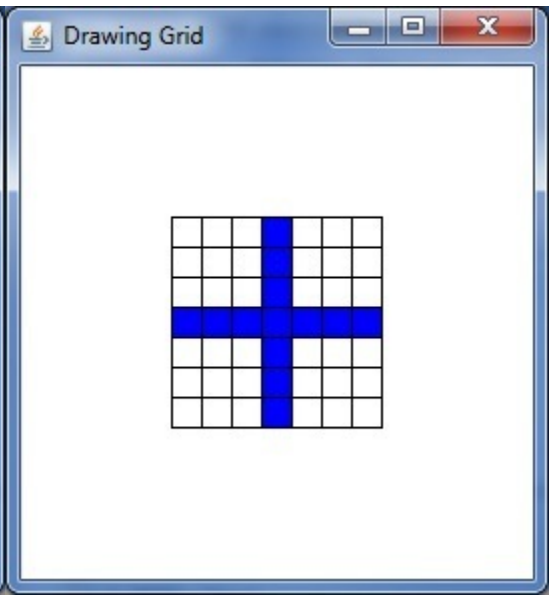
[See the shapes you must draw on the back page!]

This is what the shapes should look like when drawn on a SquareGrid of size 7:

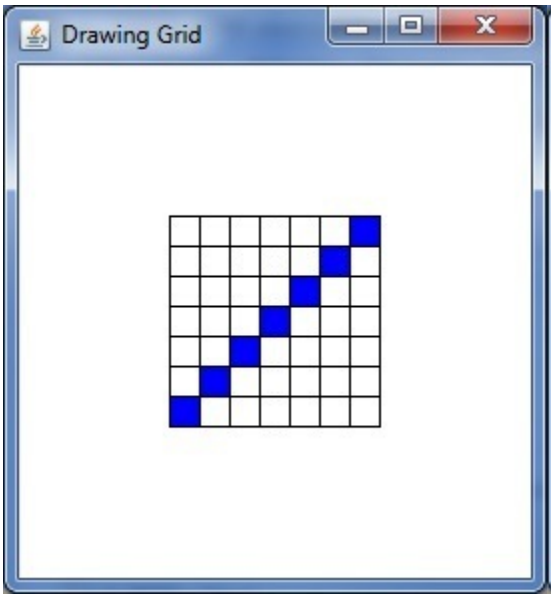
Minus:



Plus:



Divide:



Multiply: