CSCI 241 Assignment 8 Drawing Patterns

Due: Thursday, April 15, 2021 *at 11:59 p.m.*

For 3 points, email your instructor the code for the print() method by 11:59 p.m. on Tuesday, April 13, 2021.

Objective: Use static (class) methods to modify and print a 2-dimensional array of characters with different patterns.

Description:

Begin your assignment by creating a new BlueJ project named **`Assign8Patterns`**. Add a class named **`Patterns`.** When finished, the `main()` method in the `Patterns` class will call all the other methods in the class.

First, add these items to the class:

1. A *public class constant* to hold the character that appears in the patterns (in the examples below, it is the pound sign or hashtag (#) character. ***You can choose any character you like*** *(except the space character –I will use the pound sign/hashtag character for examples in this document).*
2. A *private class variable* to hold the number of rows and columns (which are equal) in the 2-D array that is used in the `main()` method.

Many of the methods in this class will *manipulate* a 2-dimensional array of characters. **ONLY the `print()` method should contain print statements (with this exception:** *"Running …" messages should be printed from `main()`***).** The characters in the array will be either the constant character or a space. Methods will work to create patterns in the array or to clear it out. The `print()` method can be used to print the grid declared in `main()`.

Here is a description of each method you should include:

`main()`

Uses this algorithm:

1. Call **`setDimension()`** to ask the user for an array dimension. This method is described below. It will save that value in the static variable.
2. Declare a 2-dimensional array of characters using that dimension for the number of rows and columns.
3. Call the `clear()` method to initialize the array contents to the space character.
4. Call each of the pattern-building methods below to create, then print, the revised array grid from each of the other methods. It should print them (with a heading) *in this order*: `odds(), evens(), upperLeft(), upperRight(), lowerLeft(), lowerRight(), joinedBlocks()`.
   See later pages of this document for sample output.
   Each time a pattern is created, *you decide* if the array needs to be cleared before building the pattern by calling the **`clear()`** method (described below). Once each pattern is built, call the **`print()`** method (described below) to display it in the terminal window. *This means that the `print()` method will be called a total of 7 times* (ONLY FROM `main()`). **DO NOT call `print()` from any other method!**

BlueJ hint: As you test your code, you may need to click Options-Unlimited Buffering in the BlueJ terminal window to see all results.

`void setDimension ()`

Takes no parameters and returns nothing. It first declares a keyboard `Scanner`. It will ask the user for one value to save in the corresponding static variable (described on page 1). This value is the dimension (number of rows and columns) for the character array. The dimension must be an odd number between 5 and 15, both inclusive. Keep asking the user for this number until it fits these criteria.

`print(char [][])`

For 3 points, email your instructor the code for this method by 11:59 p.m. on Tuesday, 4/13.

Takes a 2-dimensional character array as a parameter and returns nothing. Its job is to print the array contents (from the parameter) in a neat format. For example, if the array has dimensions 5 x 5, and has the pound sign/hashtag symbol (#) in every position, this method will print:

```
-----------
 # # # # #
 # # # # #
 # # # # #
 # # # # #
 # # # # #
-----------
```

Note that the method's printing *must* contain the top and bottom bordering – symbols. The number of columns in the array will determine how many hyphens to print above and below the array contents. The method should print one space **before** each of the array's characters.

`clear(char [][])`

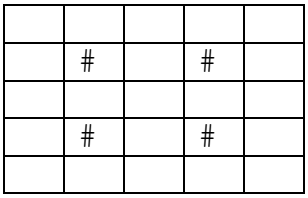
Takes a 2-dimensional character array as a parameter and returns nothing. Its job is to place the space character in each location in the array.

Write 7 additional methods that manipulate the contents of the 2-D array of characters sent in the parameter. All methods have the same parameter (`char [][]`). All have a `void` return type.

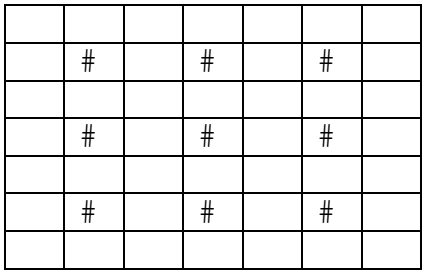
As you write these methods, you can do some of them the "hard" way, and some the "easy" way. If an already-written method does part of what you need to do, go ahead and call it from your new method! You can call one method from another just as easily as calling a method from `main()`. Most methods need a nested loop. Some might need no loops at all! You decide...

The description of each of these methods shows the method signature and what should result for a 5x5 array on the left, and a 7x7 array on the right. Remember, the array may have up to size 15 for its dimensions.

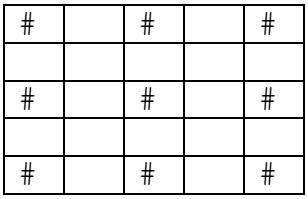
odds (char[][])



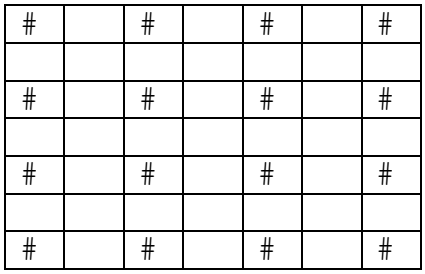
odds (char[][])



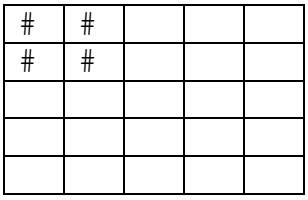
evens (char[][])



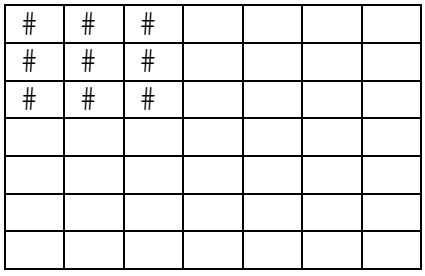
evens (char[][])



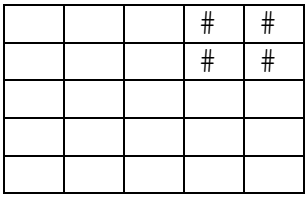
upperLeft (char[][])



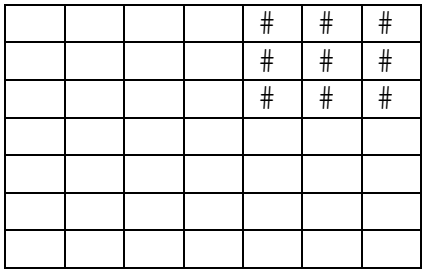
upperLeft (char[][])



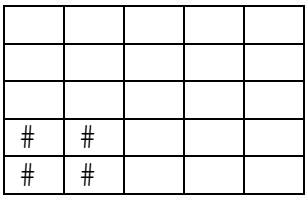
upperRight (char[][])



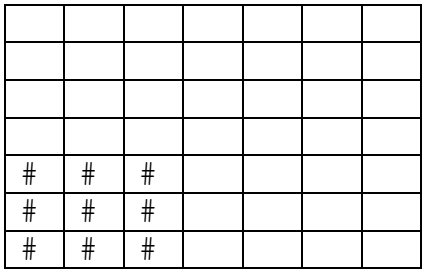
upperRight (char[][])



lowerLeft (char[][])



lowerLeft (char[][])



lowerRight (char[][])

			#	#
			#	#

lowerRight (char[][])

				#	#	#
				#	#	#
				#	#	#

joinedBlocks (char[][])

#	#		#	#
#	#		#	#
		#		
#	#		#	#
#	#		#	#

joinedBlocks (char[][])

#	#	#		#	#	#
#	#	#		#	#	#
#	#	#		#	#	#
			#			
#	#	#		#	#	#
#	#	#		#	#	#
#	#	#		#	#	#

Make sure to document all methods and their tricky code sections. A finished version of the output for a 9x9 array is on the last 2 pages of this document. *Your output should be an exact match!*

Submission Requirements

1. **Electronic:** Use SSH (Secure Shell) to put your BlueJ project into your CS lab account. Use Remote Desktop Connection to get your lab desktop so you can use the **BlueJ** submission tool to electronically submit your **`Assign8Patterns`** project from your CS lab account.
2. **Canvas copy:** Submit your `Patterns.java` file to the associated assignment entry in Canvas. I can make notes for you there instead of on paper.

Grading Criteria

Your program *must compile without errors to be accepted*. *Make sure your BlueJ project name is also correct!*

The following criteria will be used to evaluate your program:

Early points for emailing print() method to instructor	3 pts
Adherence to style and design conventions (identifier names, indentation, etc.)	2 pts
Comments	4 pts
Organization and efficiency of methods	3 pts
Correct program behavior	18 pts
Total points:	**30 pts**

See next 2 pages for an example of expected output for an entered value of 9:

As usual, data entered from the keyboard is shown **bold and underlined** to distinguish it from what the program prints:

```
Enter grid dimension (odd between 5 and 15): 9
Running odds()
------------------

   #   #   #   #

   #   #   #   #

   #   #   #   #

   #   #   #   #

------------------
Running evens()
------------------
 #   #   #   #   #

 #   #   #   #   #

 #   #   #   #   #

 #   #   #   #   #

 #   #   #   #   #
------------------
Running upperLeft()
------------------
 # # # #
 # # # #
 # # # #
 # # # #




------------------
Running upperRight()
------------------
            # # # #
            # # # #
            # # # #
            # # # #
```

```
--------------------
Running lowerLeft()
--------------------




 # # # #
 # # # #
 # # # #
 # # # #
--------------------
Running lowerRight()
--------------------




           # # # #
           # # # #
           # # # #
           # # # #
--------------------
Running joinedBlocks()
--------------------
 # # # #   # # # #
 # # # #   # # # #
 # # # #   # # # #
 # # # #   # # # #
         #
 # # # #   # # # #
 # # # #   # # # #
 # # # #   # # # #
 # # # #   # # # #
--------------------
```