Program 19 – Have a penny, leave a penny….

In the game of Penny Pitch, a two dimensional board is laid out as follows:

```
1 1 1 1 1
1 2 2 2 1
1 2 3 2 1
1 2 2 2 1
1 1 1 1 1
```

A player "tosses" five pennies on the board, trying for the number with the highest value. At the end of the game, the sum total of the tosses is returned. Develop a program that plays this game. The program should display the board, the toss number, the total score, and perform the following steps each time the user presses Enter:

- Generate two random numbers for the row and column of the toss.
- Add the number at this position to the runing total.
- Display the board, replacing the numbers with * where the pennies land.

(*Hint:* Use a two dimensional array of Square objects (a different class) for this problem. Each square contains a value number like those shown, and a Boolean flag that indicates whether or not a penny has landed on that square [see next page])

Sample Output:

```
Welcome to penny toss.

Press <enter> to throw a penny/continue.

Throw #1 of 5

* 1 1 1 1
1 2 2 2 1
1 2 3 2 1
1 2 2 2 1
1 1 1 1 1

You got a 1.
Your total score so far is: 1
Press <enter> to throw a penny/continue.

Throw #2 of 5

* 1 1 1 1
1 2 2 2 1
1 2 3 2 1
1 * 2 2 1
1 1 1 1 1

You got a 2.
Your total score so far is: 3
Press <enter> to throw a penny/continue.

Throw #3 of 5

* 1 1 1 1
1 * 2 2 1
1 2 3 2 1
1 * 2 2 1
1 1 1 1 1

You got a 2.
Your total score so far is: 5
Press <enter> to throw a penny/continue.

Throw #4 of 5

* 1 1 1 1
1 * 2 2 1
1 2 3 2 1
1 * 2 2 1
1 1 1 1 1

You got a 2.
Your total score so far is: 7
Press <enter> to throw a penny/continue.

Throw #5 of 5

* 1 1 1 1
1 * 2 2 1
1 2 * 2 1
1 * 2 2 1
1 1 1 1 1

You got a 3.

Your final score was 10.
Would you like to try again? (y/n)
```

Penny landed on an occupied square. Just add in the value.

Class SquareObject

Private instance variables

int *squareValue*	// holds value of current square
boolean *present*	// tells if square holds a penny

Methods

Default constructor	// sets *squareValue* to 1 and *present* to false
Constructor with parameter	// sets *squareValue* to parameter, *present* to false
getSquareValue	// returns value of *squareValue*
setSquareValue	// sets value of *squareValue*
hasPenny	// returns value of *present*
setPenny	// sets value of *present*

Class Program19

Methods

main	// main loop to get input, make random numbers
	// set the state of individual squares, etc.
displayBoard *	// displays the received board
resetBoard *	// fills/resets the received board by filling it with
	// square objects of the correct value/state

*Note: because main is a static method (more on this later) and methods/variables declared in the Program19 class that the main method wishes to access must either be declared locally or declared as static as well.

For a couple of extra points (2-3), ask the user for the size of the grid, and the number of pennies (tries) to toss. Follow the same pattern for determining the grid layout.

Ex: a grid that is 7x7 would like this:

```
1 1 1 1 1 1 1
1 2 2 2 2 2 1
1 2 3 3 3 2 1
1 2 3 4 3 2 1
1 2 3 3 3 2 1
1 2 2 2 2 2 1
1 1 1 1 1 1 1
```