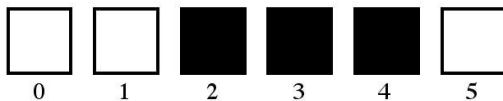


# ITP 109: Introduction to Java Programming

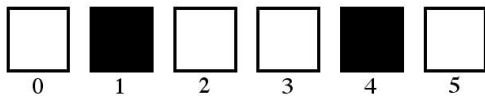
## Assignment 07 Using arrays

In this assignment you will create a program from scratch; a one-dimensional game of Lights Out. Briefly, the game consists of a row of lights that can be either on or off. By choosing a particular light, you switch its state and the state of its neighbors. That is, you turn the chosen light from off to on or from on to off, and you turn each neighbor from off to on or from on to off, depending on their initial states. The game is won when all of the lights are turned off.

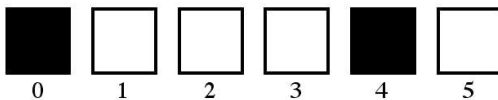
For example, if there were 6 lights in the game in the following state (dark squares indicate "off", light squares indicate "on"):



and the user selected light number 2, then the resulting state would be :



If the user then selected light number 0, the resulting state would be :



Doing so you will practice...

- Creating methods
- Using arrays
- Testing your code

## Procedure

**Step 1:** Open BlueJ and create a new project called **Assignment07-username**, where username is your USC username (your email address without the @usc.edu). You will have one file called **LightsOut** that contains all the logic for the game.

**Game Overview:** Your program should first prompt the user for a number of lights. This must be between 3 and 15 (inclusive). If it is not in that range, your program should enter a small loop in which it continues to prompt the user for an integer in the valid range until one is typed. You may assume the user will always input an integer.

Your program should create an array of the specified length and go through each value in the array and randomly set it to "on" or "off". This randomization is for each value separately -- we want a random pattern of lights to begin with.

You should then print out the pattern of lights so that each "on" light is a 4x4 block of \*'s and each "off" light is a 4x4 block of spaces. Each light should be separated by vertical bars (|). After printing the lights, you should include numbers beneath each light, starting from 0.

A full-sized, fifteen-light initial configuration might look like this:

```
|****|****|   |   |   |****|   |****|****|****|   |   |   |   |****|
|****|****|   |   |   |****|   |****|****|****|   |   |   |   |****|
|****|****|   |   |   |****|   |****|****|****|   |   |   |   |****|
|****|****|   |   |   |****|   |****|****|****|   |   |   |   |****|
  0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
```

Then, you should ask the user what light they would like to select (it must be an integer). You should check to be sure that the light selected is actually valid for the current game -- if not, you should enter a small loop and continue prompting the user until a correct input is typed in. See the run below as an example. (**Also note:** For *some* numbers of lights and *some* initial states, the Lights Out puzzle isn't even solvable. So there needs to be an option to let the user quit if they can't get to a solution!)

Once a valid light number is input, your program should make the appropriate changes to the light array, and then continue by printing out the new array and prompting the user again. When the user wins the game by turning off all of the lights, the program should stop, print a congratulatory message, and quit. There is no need to prompt the user to play again; in this case, we'll let the user rerun the program if they would like to.

Here is a sample run of my program as a guide to an appropriate interface. The user's input is underlined:

```
Welcome to Lights Out!
How many lights would you like to have (3-15)? 8
```

```
|****|****|   |****|****|   |   |   |
|****|****|   |****|****|   |   |   |
|****|****|   |****|****|   |   |   |
|****|****|   |****|****|   |   |   |
  0    1    2    3    4    5    6    7
```

```
Which light do you select (or -1 to quit)? 1
```

```
|   |   |****|****|****|   |   |   |
|   |   |****|****|****|   |   |   |
|   |   |****|****|****|   |   |   |
|   |   |****|****|****|   |   |   |
  0    1    2    3    4    5    6    7
```

```
Which light do you select (or -1 to quit)? 3
```

```
|   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |
  0    1    2    3    4    5    6    7
```

You win! Well done.

## LightsOut Specifics

- **Variables:**
  - The boolean array of lights
  - A boolean that represents if the game is won or not
  - *Optional:*
    - 2 constants (boolean ON = true and boolean OFF = false). I find it helpful to think of ON and OFF instead of true/false, but it is up to you.
    - Scanner for user input (optional, can be a local variable or instance variable)
- **Methods:** each individual task of the game belongs in a method. You must have at least the following methods in your program (in no particular order):
  - `public void start()` – this method controls the whole flow of the game. It is the only public method.
  - `private int getNumberOfLights()` – this method is responsible for prompting the user for the number of lights he/she wants in the games. It will only return a valid number, [3-15]
  - `private void initializeLights()` – this method will initialize each spot in the array to true or false. Note that the Random class has a `nextBoolean()` method that can be used.
  - `private void checkWin()` – this method updates the boolean instance variable if needed and displays a message to the user
  - `private boolean isValid(int light)` – this method returns true if a given light is valid for the game (that is, it's between valid array indices: 0 to number of lights)
  - `private int getLightNum()` – this method handles the user input for which light they want to choose. It only returns a valid light number or -1.
  - `private void changeLights(int lightNum)` – this method controls the flow of changing lights, based on the `lightNum`. It switches the light at spot `lightNum`, and then makes sure neighboring lights are valid before switching their colors.
    - **Note:** At the end of this method or in the start method after the call to `changeLights`, you should check to see if the game is won after these lights are changed.
  - `private void printLights()` – this method “pretty” prints the lights according to the printing specifications above
- Make sure to **comment** your code and use good style like proper indentation.

## Deliverables

- A compressed **Assignment07-username** folder containing two files (Location.java and Quizlet.java). It must be submitted through Blackboard.  
[Here are the instructions for submission](#)
  - a) Navigate to your project folder.
  - b) Create a compressed folder with all your code.
  - c) Rename the zip file as follows: **assignment07\_lastname\_firstname**
  - d) Upload zip file to Blackboard site for our course.

Please note, if you do not follow the submission instructions, your submission **will not be graded and/or will be counted as late**. Please make sure you have named your files appropriately.

## Sample output

Welcome to Lights Out!

How many lights would you like to have (3-15)? 10

```
|      |****|      |      |****|      |      |      |      |      |
|      |****|      |      |****|      |      |      |      |      |
|      |****|      |      |****|      |      |      |      |      |
|      |****|      |      |****|      |      |      |      |      |
  0      1      2      3      4      5      6      7      8      9
```

Which light do you select (or -1 to quit)? 2

```
|      |      |****|****|****|      |      |      |      |      |
|      |      |****|****|****|      |      |      |      |      |
|      |      |****|****|****|      |      |      |      |      |
|      |      |****|****|****|      |      |      |      |      |
  0      1      2      3      4      5      6      7      8      9
```

Which light do you select (or -1 to quit)? 1

```
|****|****|      |****|****|      |      |      |      |      |
|****|****|      |****|****|      |      |      |      |      |
|****|****|      |****|****|      |      |      |      |      |
|****|****|      |****|****|      |      |      |      |      |
  0      1      2      3      4      5      6      7      8      9
```

Which light do you select (or -1 to quit)? 1

```
|      |      |****|****|****|      |      |      |      |      |
|      |      |****|****|****|      |      |      |      |      |
|      |      |****|****|****|      |      |      |      |      |
|      |      |****|****|****|      |      |      |      |      |
  0      1      2      3      4      5      6      7      8      9
```

Which light do you select (or -1 to quit)? 3

```
|      |      |      |      |      |      |      |      |      |
|      |      |      |      |      |      |      |      |      |
|      |      |      |      |      |      |      |      |      |
|      |      |      |      |      |      |      |      |      |
  0      1      2      3      4      5      6      7      8      9
```

You win! Well done.

## Grading

Item	Points
<b>getNumberOfLights returns 3-15</b>	4
<b>initializeLights randomly sets array's values</b>	4
<b>Start method sets up game properly</b>	3
<b>Start method loops for valid game play</b>	5
<b>checkWin properly maintain game won status</b>	5
<b>isValid</b>	3
<b>getLightNum</b>	3
<b>Changelights: change light and it's neighbors properly</b>	5
<b>printLights</b>	5
<b>Comments, style, and proper submission</b>	5
<b>Total</b>	<b>42</b>

Remember that **looking at, sharing, or copying anyone else's code**, is considered [cheating](#). If you need help with something, ask in class, ask one of the instructors during office hours, ask a TA in office hours, ask on [Piazza](#), or ask a peer to **verbally** explain a **concept** to you.