# COSC 1437   Programming Lab & Assignment 01   (Spring 2015)

Topics

- practicing **for** loop structures with Dot puzzles
- Hyuk Cho adapted the original content from
  http://www.bowdoin.edu/~ltoma/teaching/cs210/fall10/Labs/Lab1-Loops/lab1.html

Overview

In this laboratory & assignment, you will draw a series of 16 dot patterns using nested for loops. You may think of the dot patterns as geometric puzzles to be solved using loop structures.
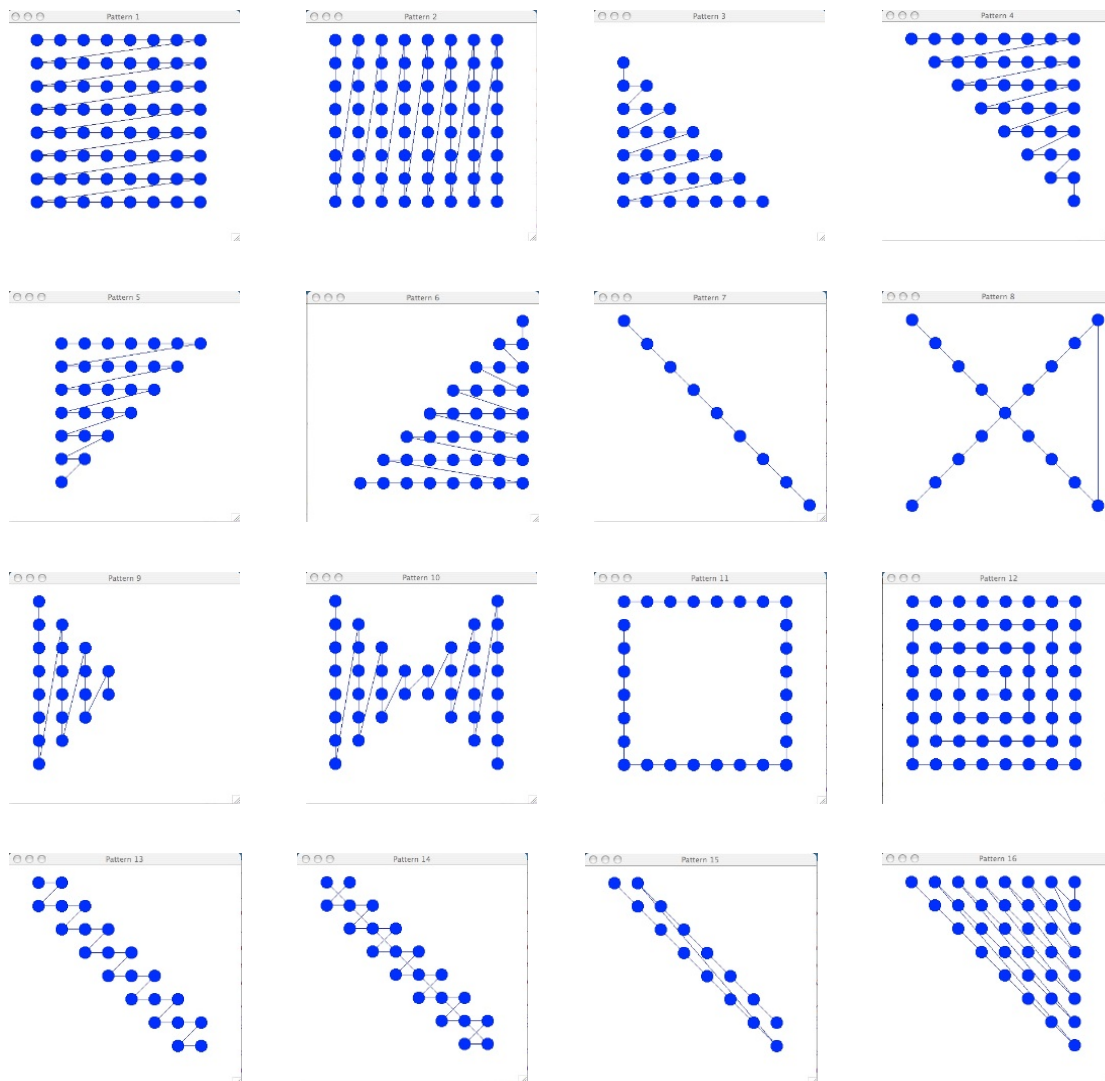
Project Files

To start on this lab download a file called `Dots.zip` on your working directory. Double-click on the icon and it will expand into a folder called `Dots`. Look inside this folder; then, you will find the following five files: `Dots.pdf`, `SolutionDots.class`, `Dots.java`, `DrawingBoard.java`, and `MySolutionDots.class`.

Compiling & Running: Assume you are using TextPad as your IDE or use java compiler and java virtual machine in command-line as follows:

- Using TextPad: `Read Dots.java` onto TextPad; then, compile and run the code through "Tools→External Tools→Compile Java" or "Tools→External Tools→Run Java Applications", respectively.
- Using java compiler and java virtual machine in command-line: Open the command line by typing ***cmd*** in "search programs and files" window; then, go into your working directory, compile, and run it. For example,

```
> cd Desktop\COSC1437_Spring15\Lab_Assignment_Spirng15\Dots\
> javac Dots.java
> java Dots
```

These are Windows commands. The first command goes to your working directory (i.e., folder). The second command compiles the `.java` files and creates `.class` files. You may notice that `SolutionDots.class` does not have the source code (i.e., `.java`. Note that, for your information, `MySolutionDots.class` is given to show you the expected solution (under the title "Pattern n") beside "Your Pattern n" window.

Getting started

When you run the program you can enter numbers for the various patterns you have to draw. The right window will draw what your pattern is supposed to look like (i.e., the expected patterns), while the left window will be blank except the first pattern (i.e., Pattern 1).

In `Dots.java`, you will find that the shape of Pattern 1 is controlled by the following nested loop structure in the first block of the switch statement in `drawPattern`:

```
switch (pattern) {
case 1:
      for (row = 0; row < ROWS; row++)
           for (col = 0; col < COLS; col++)
                board.bigDot(row, col);
      break;
```

As this nested loop executes, method `bigDot` automatically connects each dot to the next dot that is drawn. You may ask: "Why are most of the connections horizontal lines?" The answer lies in the order in which the loops are nested. The row loop is the outer loop and the column loop is the inner loop. This means that, for each value of the row index, an entire cycle of the column

loop must be executed. Therefore, for each row, the inner column loop draws a dot in column 0, then in column 1, and so on through column 7. This explains why the dots in each row are connected by a sequence of horizontal lines. When a given row is finished the last dot drawn is in column 7. The next row begins with a dot in column 0. This explains the diagonal lines from upper right to lower left which connect successive rows.

The method `bigDot(row, col)` arranges the details of drawing the dots and making the lines which connect successive dots. Since these details are not the main focus of this exercise, we will not discuss them here.

Your Task

Your task is to draw the remaining 15 dot puzzles by using appropriate loop structures. You should insert the code for the remaining 15 patterns as blocks in the switch statement of `drawPattern`. See the location in the file marked with the comment:

`Enter additional pattern solutions below`

When your solutions are entered in the switch statement, the driver routine `userLoop` will enable you to interactively test all of the patterns and compare them against the solutions.

Each puzzle requires giving a new twist to the loop structures. For example, in puzzles 3 to 6, you must figure out how to replace the constants 0 and 8 in the inner loop by expressions which depend on the row index. In some of the later puzzles, you will need a sequence of loop constructs since a single nested loop structure will not be sufficient. In a few cases, you will need an extra loop variable besides row and col. However, **try to use as few loops as possible, and no additional instructions besides for loops.**

Grading scheme

Points will be awarded as follows. Therefore, the maximum score will be 100 points.
- Each pattern is worth 6 points.   (➔ 90 = 15 * 6)
- There is a 10 point bonus if all patterns are correct.   (➔ 10)
- Therefore, if you correctly generate only 10 patterns (out of 15 patterns), you will be given 60 points (60 = 10 * 6 + 0).

Thing to submit

- The TA will check your file using the lab computers. Therefore, just in case you are using different programming environment, please make sure your program is successfully compiled and run on the lab computers in AB1#209 or AB1#211. If your program is not compiled in the lab computers, no points will be given.
- Submit only one java program (i.e., "**Dots.java**") via Blackboard before the next lab.