

CS231 Fall 2013 Lecture 7

Stephanie R Taylor

September 18, 2013

1 Administrative Topics

- Sign up for csstu if you aren't already on it.

2 PSVM and Command line arguments

- *public static void main*
- By now you should understand pretty much everything in the header of the main method:

```
public static void main(String [] args)
```

- What does that line tell you?
- It is a public method so that anyone can call it.
- It is static so you don't need to create an object to call it.
- It has return type void so it doesn't return anything.
- It takes an array of String objects as its parameter. That array has the name "args".
- Until now, we've ignored the args parameter, but you will need it for the project this week. It is used as a way of passing in values on the command line.

- Let's look at a simple example:

```
class CLTester {
    public static void main(String[] args) {
        System.out.println(args[0] + args[1]);
    }
}
```

- Now let's execute this via "java CLTester Hi There" [Do it]
- But what if you wanted to enter two numbers on the command line and then add them? Will the code work if I run "java CLTester 3 4"? [Do it]

```
class CLTester {
    public static void main(String[] args) {
        System.out.println(args[0] + args[1]);
    }
}
```

- The args are treated as strings, even if you run "java CLTester 3 4". To add them as integers, you need to convert the strings to integers. Here's how you can do it:

```
class CLTester {
    public static void main(String[] args) {
        int arg0 = Integer.parseInt(args[0]);
        int arg1 = Integer.parseInt(args[1]);
        System.out.println(arg0 + arg1);
    }
}
```

3 2D Arrays

- What if you want to represent something that is naturally 2 dimensional, such as a chess board. One way is to have arrays of arrays:

```
int [][] matrix = new int [5][4];
```

This creates an array with 5 slots, each of which is an array of length 4 [draw a picture with pointers]

- Let's play with this matrix. What do the following lines of code do?

```

matrix[0][1] = 3;
for(int j = 0; j < 4; j++)
    matrix[0][j] = 1;
for(int i = 0; i < matrix.length; i++)
    for(int j = 0; j < matrix[0].length; j++)
        matrix[i][j] = i+j;

```

- We don't need all inner arrays to be the same length:

```

int [][] m = new int [2][];
m[0] = new int [] {1,2,3};
m[1] = new int [] {1,2};

```

4 Project 2 Advice

The goal of the project is to become comfortable with 2D arrays while writing code to simulate John Conway's game of life.

Here are some pointers:

- From now on, the instructions will allow for more independence and will not necessarily contain instructions to make every method you need to solve the problem. So, think about the design as a whole before you begin coding.
- The game of life is played step by step. At each step, each cell changes its value. The cell makes its decision regarding that change beaded on the values of its neighboring cells at the previous time step. Thus, it is imperative that you have two copies of the board, one that allows each cell to find its neighbors and one so that it can record its decision.
- The classes described in the instructions follow a sort of hierarchy:
 1. LifeSimulation: this is top-level code that drives the simulation. A LifeSimulation contains a Landscape.
 2. Landscape: this is the object that manages the 2D grid of Cells.
 3. Cell: the entity at each location in the grid.