CMSC131

Grids/Matrices and Nesting of Loops

Grid/Matrix Metaphor

The following (and more) all have something in common:

– Spreadsheet
– Databases
– Graph Paper
– Computer Screen
– Command-line window or sheet of paper in a typewriter (when using a fixed-width font)

They can all be thought of in terms of rows and columns in the form of a basic two-dimensional structure, sometimes called a matrix or a grid.
Interacting with Grids/Matrices

While the metaphor or rows and columns can be applied to many things, the nature of the item can restrict how we interact with it.

– In a spreadsheet, you can enter a row/col pair and jump right to that position.
– On a computer screen, programs can logically specify an x,y coordinate to place something there, but on-screen the entire image itself is refreshed using a fixed pattern.
– With a sheet of paper in a typewriter or a command-line window, you often have to access it line by line, left to right with no way to go back.

Nested Loops

It is very common to nest for loops and sometimes to nest while loops.

– You saw an example application of nesting of loops in lab today, and will see this nesting in a wide variety of contexts throughout your studies and career.

The nesting of for loops aligns very well with things that can be thought of as grid-like.
What would this look like?

```java
public class NestedForLoopsV1 {
    public static void main(String[] args) {
        for (int row=1; row<=5; row++) {
            for (int col=1; col<=5; col++) {
                System.out.print("["+row + "," + col + "]");
            }
            System.out.println();
        }
    }
}
```

Write the output on a sheet of paper or type it into a simple text editor, then show it to your neighbor and compare.

What change would this cause?

```java
public class NestedForLoopsV2 {
    public static void main(String[] args) {
        for (int row=1; row<=5; row++) {
            for (int col=1; col<=row; col++) {
                System.out.print("["+row + "," + col + "]");
            }
            System.out.println();
        }
    }
}
```

Write the output on a sheet of paper or type it into a simple text editor, then show it to your neighbor and compare.
public class NestedForLoopsV3 {
    public static void main(String[] args) {
        for (int row=1; row<=5; row++) {
            for (int col=row; col<=5; col++) {
                System.out.print("["+row + "," + col + "]");
            }
            System.out.println();
        }
    }
}

Write the output on a sheet of paper or type it into a simple text editor, then show it to your neighbor and compare.

Would this produce the right look?

for (int row=1; row<=5; row++) {
    for (int col=1; col<row; col++) {
        System.out.print(" ");
    }
    for (int col=row; col<=5; col++) {
        System.out.print("["+row + "," + col + "]");
    }
    System.out.println();
}
while rather than for

```java
int row = 1;
while (row <= 5) {
    int col = 1;
    while (col <= row) {
        System.out.print(row + ""," + col + " ");
        col = col + 1;
    }
    System.out.println();
    row = row + 1;
}
```

Notice that here the loop control variable needs to be dealt with in several places rather than all on one line as with the for loop. This is seen as a weaker style than for loops in this type of context.

More to come…

The nesting of coding elements such as conditionals, iteration, etc. and the computational thinking approaches related to them are topics that we will explore more and more as the semester progresses in a variety of contexts.