Two-dimensional Array Processing CS 205

Overview

The primary purpose of this program is to gain experience processing a two-dimensional array.

Requirements

- create a 10 x 10 integer array named **table**.
- Use a nested loop to initialize each array element to a random number between 0 and 99 inclusively.
- After the nested loop above has completed, use a second nested loop to locate the highest number in the array. Keep track of this number as well as its row and column index.
- After the second nested loop has completed, use a third nested loop to display the array in a table format. Format the output so that each number output is right aligned (default) in at least a field width of 3.
- Finally, display the highest number and its location in the exact format shown in the sample runs.

- Other notes
  - Use three separate sets of nested loops as described above. One could shorten this into one nested loop, but one of the goals is to gain more experience with nested loops and two-dimensional arrays.
  - The only documentation required is your name as author just above the class.
  - Dividing code into separate methods that each perform a single task is optional: All the code may be placed into Main.
  - **Remember to use named constants** in place of numbers such as 10 and 99 (or any number other than 0, 1, or 2 that is not in a formula nor used for formatting).

Two Sample Runs

```
 23 51 11 85 61 71 25 33 13 83
 46 33 46 28 91 46 76 88 29 34
 11 59 90 71 55 40 98 71 64 77
 92 48 35 99 56 69 85 67 51 75
 54 48 23 65 30 68  0 17  9 84
 61 27 32 46 46 42 37 62  3 86
 56 17 35 10 12  9 13 90 81 84
  3 73 63 47 66  3  9 17 54 76
  9  0 27 88 38 83  9 70 78 71
 69 57 38 95 12 78 40  4 54 88

The highest value located at [3][3] is 99
```

```
 83 62 93 17 29 39 44 21 95 83
 16 71  5 39 61  4 22  9 16 98
 44 48 24 69 25 31 77 56 97 94
 67 67 30 33 74 28 11 22 41 16
 34 85 46 48 96 96 66 94 73 68
 97 42 45 85  1 63 19 93 92 22
 23 82 65  0 76 89 63 86 70 71
  9 65 79 73 51 23 76 22  4 59
 21 21 56 17 22 10 80 49 57  5
 58  8  2 61 74 43 75 69 37 48

The highest value located at [1][9] is 98
```

Submission

- Before class: Print and upload the source code to the designated online drop box.
- Beginning of class: Hand in the source code and staple if needed.