Create a battleship board and manually populate it with “boats”. (I.e. hard code the boats into valid positions on the board). Print out the board showing the boats.

A battleship board is a 10 X 10 square. The boats are:

Aircraft Carrier: 5 long

Battleship: 4 long

Destroyer: 3 long

Submarine: 3 long

Patrol Boat: 2 long

You don’t have to know what boat is at what spot as long as you can tell that a boat is there, like in the real game

Harder – If time allows:

1. Place the boats in random locations. (vertically or horizontally) - BUT
   1. Don’t allow boats to overlap – i.e. no two boats can occupy the same spot on the board.
   2. Don’t allow boats to hang off the edge of the board.

Hardest – (if you can) - Allow user to play game. Use Console.ReadLine and Convert.ToInt32 to get row and column and then somehow mark hits and determine if game is over.

Hints:

Represent the board as a 2 dimensional int array ([10][10] or [10,10]) (or char array)

Represent empty spaces as 0

Represent spaces that are filled by a boat by a 1 (or even better, a different number or letter for each boat, if you want to do that)

Randomly switching from horizontal vs vertical is just like flipping heads and tails from our coin flip HW

Consider making an array of boats (hint: the array of boats can just be an array of ints (not even a multi-array)) – not required but might result in less duplicate code.

It will probably be easier to follow if you create some methods to do some of the work and call them rather than coding everything in main.

To keep boats from going off the edge – restrict the highest possible number you can pick to be one that allows that particular boat to be placed.