Java vs. C#

1) *using System;* // *System* is a *namespace* in C#
   using System vs. *import*

2) public class MyFirstC { ← *Doesn't need to match file name*

3) public static void Main (String [ ] args) {
   Main() vs. main ()
   …only required if using CMD Line Args (…same is also true in Java)
   …Compiles fine w/*Main(String [] args)* or *Main(string [] args)*

string vs String: (string is the actual data type…)

- In C#, *string* is an *alias* for *System.String*. As long as "*using System*" is included, *String* will compile to the same internal code as string

4) **Comments / Data types** same except for **bool** (vs boolean)

5) **Constants** declared with *const* .vs *final* keyword

6) **Arithmetic operators** are the same (+ - / * %)
   - *Conditionals* (if-else, switch) and *Loops* are the same
   - Switch requires '*break*' after each case to compile

7) **Output** - *Console.WriteLine*/*Write* …C# ***methods*** start w/capital (vs. *System.out.println/print*)

8) **Input -** *Console.ReadLine()*, ***Convert****.ToInt32* (string)
   Convert*.ToDouble* (string)
   (vs. System.in/Scanner, nextLine(), parseInt/parseDouble)

Regular **int***: 32-bits (4 bytes) … 2^{31} = +/- 2.1 billion - Convert.**ToInt32**
short (int): 16-bits (2 bytes) … 2^{15} = +/- 32,767 - Convert.**ToInt16**
long (int): 64-bits (8 bytes) … 2^{63} = +/- 9.2E18 - Convert.**ToInt64**

**int* is an *alias* for *System.Int32* / *double* is an *alias* for *System.Double*
…compiles to same code

9) Character Conversion / Input

Method 1: Convert entire string to char

char ch = *Convert.ToChar(response)*;

…throws exception if string (*name*) is not exactly one char

Method 2: (**Better**) Strings in C# can be referenced like a *char array*

…pull off first letter of the string

char ch = response[0];

10) Converting a *string*(s) into a *Character Array*

(String and Character Array are 2 different data types)

char [ ] charArray = s.ToCharArray();

---

11) 1D Arrays declared the same as in Java

2D Arrays (**Java**): int [][] grid = new int [3][4];

2D Arrays (**C#**): int [,] grid = new int [3,4];

[0,0]			**[0,3]**
		[1,2]	
	[2,1]		**[2,3]**

METHOD GetLength (**not** *data property* Length)

grid.GetLength(0) = dimension 1 … returns number of *rows*

grid.GetLength(1) = dimension 2 … returns num of *columns*
