

Slide 1

Administrivia

- Reminder: Quiz 2 Tuesday. Likely topics are arrays, uses of `String` class.
- Reminder: Homework 2 design due today at 11:59pm, code next Tuesday. Grades and comments on Homework 1 code coming soon — but if you got the starter game when you ran your code, you almost certainly will get full credit.
- Sample solutions for quizzes available on the Web shortly after class.

Slide 2

A Little About Arrays in Java

- Arrays are objects — unlike in C/C++, where they're basically pointers.
- Declaring (references to) arrays — denote by putting brackets after type.
- Creating arrays — use `new`, e.g.,

```
new int[10]
```

```
new String[n]
```

(Remember that the second one only creates *references*.)
- All arrays have `length` variable.
- Otherwise, syntax is same as C/C++; indices start at 0.
- Java runtime does automatic bounds-checking — unlike in C/C++, get `ArrayBoundsException` rather than random problems.

Slide 3

Multidimensional Arrays

- “Arrays of arrays”, e.g.,

```
int[][] x = new int[10][100];
```


declares an array of 10 arrays of 100 ints.
- Reference elements with row, column indices, e.g.,

```
x[row][col] = 10;
```
- Both dimensions accessible:

```
x.length = ?
```



```
x[0].length = ?
```
- Note that order of indices (row then column) is the opposite of the “graphics convention” used in the game.

Slide 4

Homework 2 — Code

- Go back through copied code and look for things you know you want to change. E.g., you should be able to figure out how to draw something okay-for-now for your game's blocks by changing `getImage`.
- For your screen class, decide what variables you need to implement the methods in `Screen`. (You'll probably want a two-dimensional grid (array) of blocks (your block type) and a list of entities (your entity type).)

“Good Style” Tips

Slide 5

- Make methods public if needed by code that uses your class, private otherwise.
- Make variables private unless there's a good reason not to — prevents unwanted/inconsistent access.
- Try to choose good names for variables — ones that make it clear what they're for.
- Use named constants (static final variables) rather than hard-coded values.
E.g., `private static final screenWidth = 20;`
Also remember that you can get the size of an array from its `length` field (variable).
- Follow Java conventions — class names start with a capital letter, method and variable names with lower case.

“Good Style” Tips, Continued

Slide 6

- Make sure your code is indented in a consistent and readable way. (Eclipse tip: control-I cleans up indenting on whatever text is selected. So, control-A followed by control-I may be a good idea after editing that messes up indentation.)
- Remember that you're writing for three audiences: the compiler, a human reading your code, and a human reading the generated HTML documentation.

Commenting Code

- “Documentation” (javadoc) comments generally describe how someone would use your class. For examples, look at Java API, game framework, examples from class. Note that some comments for game framework describe how to “use” the class in the sense of extending or modifying it. Eventually these should disappear from your copies/versions.

Slide 7

I’ve said that for code you turn in you should provide at least a brief comment on every class and every method (and also any public variables), with the possible exception of things that are “totally obvious”. A little subjective, so getting it right will be an iterative process (you try something, I suggest changes).

- Internal comments can be helpful within the code, to clarify any tricky bits, or to improve readability. Eclipse convention seems to be to have `// TODO` comments to indicate things that will need attention at some point. Probably useful.

Homework 2, Continued

- Questions?
- (Short demo of games from previous semester(s).)

Slide 8

Minute Essay

Slide 9

- Write code to define an array of four `Strings` and fill it with data of your choice.
- Write code to define a two-by-three array of `int` and set each element to the sum of its row and column.
- If I declare an array of `MyClass` references:

```
Myclass[] objs = new MyClass[10];
```


do all the elements of `objs` have to be instances of `MyClass`, or can they be instances of some other class?

Minute Essay Answer

Slide 10

- One solution (array of `Strings`):

```
String[] s = new String[4];  
s[0] = "hello";  
/* other three lines similar */
```
- One solution (array of `ints`):

```
int[][] a = new int[2][3];  
for (int row = 0; row < a.length; ++row)  
    for (int col = 0; col < a[0].length; ++col)  
        a[row][col] = row + col;
```
- Elements of an array declared as `MyClass[]` can be instances of any "subtype" of `MyClass` — `MyClass` itself, or any subclasses. (Trick question!)