

CS231 Fall 2013 Lecture 1

Stephanie R Taylor

September 4, 2013

1 Data structures and algorithms

- One of the most important things behind medium-sized, complex programs is data management.
- This course introduces the most common data structures underlying good software design.
- Data structure: any data representation and its associated operations. Usually, the term refers to data structures that store multiple data items, such as a list of multiple floats.
- What data structures you have used? (lists, stacks).
- Efficiency matters. You want insertion, look-up, and deletion to be relatively fast.

2 Logistics

- Kyle is the lab prof. We both have office hours and will be teaching the course together, just as Dale and I and Bruce and I do for CS151.
- A week in the life of CS 231. M (lecture, lab with new project assigned, previous project due at midnight), T (work on own), W (lecture, I email you homework), R (work on own, email me homework solution and

receive reply from me), F (quiz and lecture). The lab will be staffed with CS231-prepared TAs on Sunday evening and Monday evening. The Sunday evening TA is

- Go over the syllabus, briefly.
- I will post notes, but they will not be particularly detailed.

3 Java

- There are two pages on comparing Java and Python available on the course Moodle web site. I'm giving them to you mainly for reference in case you forget the Java way.
- The best way to understand Java is to jump right in, so we'll do so.
- Java is an OO language, meaning virtually everything in it consists of objects and methods. A typical program creates some objects and has them execute their methods.
- Note there are a few non-objects (called primitives). These are integer, character, boolean, and floating point values. We'll see them a lot but you should mostly focus on the objects and classes.
- Like in Python, to create objects to do things for you, you first have to declare a class. What is a class? [a template indicating the data and methods that can be executed by that object]. Once you've declared a class, you can create a bunch of objects of that class.
- Java has thousands of built-in classes that you can use [show them the online Java API].
- But you can also create new classes of your own. In fact, creating your own new classes is almost all that you ever do in Java.
- Overall format: Java source code is stored in a simple text file with suffix ".java".

- That file cannot just include code like Python files can. Instead, such files can only contain declarations of classes, usually one declaration per file. You give the file the same name as the class declared in the file. If you declare a class named `Castle`, then the file should have the name `Castle.java`.
- Let's jump right in. Let's redo the sort of thing we did in CS151, namely let's create a class that represents a 6-sided die. Before we look at any code, what do you do with such dice? [roll them; inspect the top value].
- Objects should be able to do things for you and so we should be able to tell the die to roll itself and tell us its top value.
- The code for the `Die` class will be posted on-line.
 - Java uses curly brackets to begin and end blocks of code
 - Whitespace doesn't matter to Java (but it will to your readers!)
 - The keyword `class` is used to define a class.
 - Class names by convention start with capital letters and are camel-Case. They can be any combo of letters, digits, underscores, and dollar signs that doesn't begin with a digit and that isn't a keyword. It is recommended that you don't use the dollar sign.
 - "public" is optional but you should include it by default. It means the class is usable by every other part of the program. The other accessibility options for classes are useful only in large projects.
 - 3 parts: field declarations, constructor declarations, method declarations. Fields are for the data held by an object, constructors are called when new objects are being created, and methods are the behavior of the objects, that is, what they can do for you if you ask them nicely.
 - Each of them needs an accessibility specification. "Private" means that it can only be accessed within the class. "Public" means everything can access it. Your default should be to use private for fields and public for everything else.
 - Every field needs a type. Python variables can have values of any type (dynamically typed), but Java variables can only have values

of one type and you specify what that type is when you declare the variable (statically typed). One way to think about the difference is that in Python, data has a type but in Java data and variables both have types and if you want to assign data to a variable, the types better match.

- Constructors have names that are the same as the class name. They are mostly used to initialize values of the fields.
- Methods must include a return type in their declaration. In our example here, the first method returns an integer and the second returns nothing and so we declare it to be of type "void".
- Now let's compile and run it.
 - In the Terminal, we navigate to the appropriate directory.
 - We compile: `javac Die.java`
 - We run: `java Die`
 - But it doesn't work because we need a main function in the class.

Note: These notes were adapted from those of Dale Skrien.