15-121 – Spring, 2018
Intro to Data Structures

Lecture #3 – Strings, Random, I/O,
Loops, Problem Solving
January 18, 2018

Mark Stehlik

Outline for Today

- Strings (and the Java API)
  - StringDemo.java
- I/O (using Scanner)
- Random, Loops, Conditionals (by example)
  - PiDemo.java

Java API

- Let's look again at the StringDemo code
  - don't forget, a String variable holds a reference to a String, not a String!

- All String (and Scanner) methods can be found in the Java API

The Scanner class

- Must import java.util.Scanner (or java.util.*)
- Must declare a Scanner variable (a reference variable), create a Scanner object (via *new*), and bind it to the standard input device (*System.in*)
- Prompts are usually done with a print (vs. println)
- "reading" methods:
  - next() [reads a String – what about a char?]
  - nextInt() [reads an integer]
  - nextDouble() [reads a double]

Random numbers

- Math.random() // why no need to import Math?
  - returns a "random" double in [0,1), i.e. 0 <= r < 0.99999…
- Generate a random double between 0 and 50?
  - Math.random() * 50
- Generate a random double between 20 and 100?
  - Math.random() * 80 + 20, i.e., Math.random() * range + low
- Generate a random int between 0 to 9?
  - (int)(Math.random() * 10) // parens are important!
- Generate a random dice value (1 to 6)?
  - (int)(Math.random() * 6) + 1

Problem solving (a guided exercise)

Suppose you're on a desert island (or a desert peninsula) and you've got some time on your hands. Since it's Qatar, you've also got a bunch of pearls. And since you're a geek, you now have everything you need to calculate PI…

Static methods

- Methods provide a way to compartmentalize code (and provide for its reuse)

- A method declaration must provide a return type (*void* if no value is returned), name, parameter list, and a block (curly braces). It can also provide a visibility modifier (that defaults to *package*), and the *static* keyword which indicates it belongs to the class as opposed to an object.

- The method name and parameters (**only**) define its method signature.