Review

CSCI 1111, April 20th 2011

Understand

- Loops, Branching, conditionals
- Methods
- Arrays (pass by reference vs value)
- Strings (related methods)
- Variables (casting and scope)
- Spot compiler errors, trace programs

(No guarantee that this list is complete)

Conditionals

- && || < > <= >= == !=
- Evaluate to boolean
- ((x < 25) && (x > 20))

- Do not check doubles for equality (rounding errors)
- Strings use .equals method, not ==

Branching

- if, else if, else
- if (conditional_statement) { … }
- if (c_s) { … } else { … }
- if (c_s) { … } else if (c_s2) { … }
- if (c_s) { … } else if (c_s2) { … } else { … }
- As many else if statements as you please

Loops - for

- for (declare_i; conditional; increment_i) { … }
- Index can be int, double
- Can “count” up or down, by any increment
- Halts when conditional becomes false
  - Make sure it halts!
- for (int i = 0; i < 10; i++) { … }

Loops - while

- while (conditional) { … }
- Can do anything done with for loops
- Checks conditional at start of each iteration
  - Halts when conditional is false
  - Make sure it will halt!

Methods

- return_type name (arguments) { … }
- public static void printMin (int x, int y) { … }
- public static int factorial (int x) { … }
  - Must return an int
  - Careful with branches – easiest to have one return
- Define in class { }, but outside all other { }

Arrays

- A block of multiple variables, referenced by a index
  - Exception if incorrect index
- Should be able to copy, rotate, and reverse
- Call-by-value Vs. Call-by-reference
  - Module 12, exercise 14

Strings

- Objects, not primitives
  - Have methods such as charAt, substring, indexOf
  - Should review these methods
- Concatenation operator +
  - System.out.println(“Value: “ + x);

Variables

- Casting converts variables
  - Only needed when loss of information is possible

    int x = (int) 3.5;

    double y = x;
- Scope
  - Only accessible within the { } declared within
  - Global vs. Local
  - Local variables will obscure global

Good Luck