**Java lesson three**

1. Variables review
   1. We use mainly just four variables
      1. String – holds a string of characters, like “Hello there”
         1. Note that string literals must be enclosed in double quotation marks
      2. int – holds an integer, like -5, 0, or 17
      3. double – holds a decimal (fractional) number, like 3.5
      4. Boolean – holds true or false
   2. Can you put a word into a int variable?
   3. Can you put a number into a String variable?
   4. Converting one variable type to another
      1. Not all conversions are allowed
      2. What happens when you cast from double to int?
2. Integer division
   1. int divided by int results in int
   2. Use double instead
3. Control-of-Flow review
   1. Sequence – basically one statement after another
   2. Branch – if-else statement
   3. Loop – for statement
4. More control-of-flow
   1. switch – another branching statement
      1. Be careful – each branch must end with “break;”
      2. Optionally add “default:” as the last branch
   2. while – another looping statement
      1. Be careful – it is easy to create an infinite loop!
5. Variable scoping
   1. Basically, variables “exist” only within their enclosing curly braces
6. get and set – controlling access to your class’ variables
7. Structures - Cow can only have one of three breeds

**Practice!**

1. Have the Dog keep track of how many times it has barked.
   1. Optional extra challenge – Add a line to Main that prints out the number of barks.
   2. Optional extra challenge – Add the code necessary to be able to reset the Dog’s bark number from within Main.
2. Add a “Divide” method to Dog, similar to the “Add” method. (hint: you will need to use the data type double, not int). Call the new method from Main.
   1. Optional extra challenge – have the arguments be int, but the keep the return value as a double.
3. Add a “Catch” method to Cat, similar to the CatchMouse method. This new method will have a parameter for the type of critter to be caught as well as the number of critters to catch. The new method will have the Cat say the name and number of critter(s) caught, similar to CatchMouse. Call the Catch from Main.
   1. Optional extra challenge – use a switch statement to limit what the cat catches to a specific list of critters. Have the cat say something like “I won’t catch <critter>” when it’s not in the list.
   2. Optional extra challenge – when the critter to be caught is a mouse, call the existing CatchMouse method instead of duplicating the “I caught a…” code
   3. Optional extra challenge – do a Google search on “DRY programming pattern” and explain how Optional extra challenge #2 follows that pattern.
   4. Optional extra challenge – Notice how the “I caught a…” line is duplicated between Catch and CatchMouse, which isn’t fully DRY. Correct this. (Ask one of the instructors for a hint if you don’t know where to begin)
4. Add code to Vocalize(int Times) to check if Times is a negative number and print out “I can’t bark that” (instead of barking) when Times is negative.
   1. Optional extra challenge – rewrite Dog’s Vocalize(int Times) method to use a while loop instead of a for loop