**Java Notes Week 1**

* Programming languages
  + 3 types
    - Fully interpreted (scripted)
      * Python, JavaScript
      * We write source code and save it to hard drive
      * Interpreter converts source code to machine code, while program is running
      * Slower
    - Fully complied- source code is compiled into machine language
      * C, C++
      * We write source code and send it through a compiler which turns it into machine code
      * This happens before the program runs, unlike fully scripted
      * VEry fast
      * Executable code is different for each OS
      * High development costs
    - JAVA: best and worst of both worlds. Hybrid Language
      * Java virtual machine: works with each different operating systems
      * Source code is stored in .java files
      * Programmer pushes it through a compiler before it runs
      * Java does not compile it directly into machine code, rather byte code
      * Faster for interpreters to convert byte code to machine code
      * When we run,JDM converts bytecode to machine code
      * Bytecode will run on any computer, as JDM works with the used OS
      * Fastest
  + Python is low cost, but slower
  + There are differences in syntax, but they are all built upon the same core principles
    - Sequence (code runs in order)
    - Decision (code can branch depending on a condition)
    - Iteration (code can loop over and over)
  + Python vs Java
    - Python easier to read
    - Java we have to write more stuff
    - Python is harder to debug, but Java is structured better so its not as hard
    - Java scales better than python
    - Python has CLI, JAva doesn't
    - Python is starting to grow
    - Java is better for a very secure system, Python is becoming better at machine learning
    - Java is based on brackets not indentation, although indentation helps readability
  + How Java Works
    - JVM converts bytecode to machine code, and sends it to the OS
    - Java API is a library of code that we can use
    - JVM grabs a chunk of memory to run a program and uses it.
    - 3 parts to that memory
      * All jvm’s have a class loader, which finds the bytecode from our .class file and puts it into method area
      * The JVM always runs code at beginning of main method
      * Stack
      * Heap
      * Method
        + Where the bytecode gets loaded by JVM
    - Debugger is like a Freeze-ray and X-ray machine combined
    - Class names always start with a capital, camel case
    - CLass header tells the class name and starts the brace bracket
    - Method header has a name and a parenthesis after it like a function. Method name is camel case, no capital at start
    - Class codeblock is the code contained in the classes squiggly brackets
    - End statements with semicolon if next characters isn't an opening brace bracket
  + Variables
    - Must declare data type
    - Variables are part of memory that store data
    - Looking in the computer, it would be all 1s and 0s
    - 8 bits make up a byte
    - In 1 bit, you can store a num from -128 - 128
    - Var type is checked at compile time
    - We need to create different sizes of memory box to store different types/sizes of values
      * Byte
      * Short
      * Int
      * Long
      * Float
      * Double
      * Boolean
    - You can never declare variable type again
    - Vars whose types start with a lower case (int,double,boolean) are primitive variables (built into java)
    - Stores values for us to use
    - Strings are a var that contains a “String”
    - Numbers in a string are not actual numeric values, they are characters in a string