

Topics:	Level of Understanding		
	Huhh?	Got It	Can Teach It
• Programming Constructs			
○ Primitive data types			
▪ boolean (True/False)			
▪ int			
▪ double			
▪ void (a.k.a. ignore me)			
▪ Arrays[] (a.k.a. fixed length lists)			
• Indexing			
• two-dimensional (a.k.a. matrix)			
• array.length			
• arrays of objects			
• initialized arrays such as <code>int[] x = {1, 2, 3}</code>			
○ Declarations			
▪ Constant			
▪ Variable			
▪ Class			
▪ Interface			
▪ Method			
• Parameter			
• Return			
▪ Generics (a.k.a. <code>&lt;CLASS&gt;</code> )			
○ Primitive type casting			
▪ <code>(int)</code> (a.k.a. make me an int)			
▪ <code>(double)</code> (a.k.a. make me a double)			
○ Operators			
▪ Relational			
• <code>==</code> (a.k.a. equals)			
• <code>!=</code> (a.k.a. not equal to)			
• <code>&gt;</code> , <code>&lt;</code> , <code>&gt;=</code> , <code>&lt;=</code> (a.k.a. Greater than, Less than, Greater than or equal to, Less than or equal to, $\geq$ , and $\leq$ )			
▪ Assignment			
• <code>=</code> (a.k.a. equals)			
• <code>++</code> (a.k.a. Increment)			
• <code>--</code> (a.k.a. Decrement)			
• <code>+=</code> , <code>-=</code> , <code>/=</code> , <code>*=</code> , <code>%=</code> (a.k.a. compound assignments)			

Topics:	Level of Understanding		
	Hunh?	Got It	Can Teach It
▪ Logical			
• && (a.k.a. Logical AND)			
•    (a.k.a. Logical OR)			
• ! (a.k.a. Logical NOT)			
▪ Arithmetic			
• +, -, *, /, % (a.k.a. Add, Subtract, Multiply, Divide, Modulo or remainder)			
▪ String Operators			
• + (a.k.a. concatenation or “putting strings together”)			
• += (a.k.a. compound concatenation)			
○ Loops			
▪ for			
• for			
• for each			
• for (type x : values)			
▪ while			
○ Conditionals			
▪ if			
▪ if else			
○ Classes			
▪ Constructors			
▪ new operator			
▪ Methods			
• public			
• private			
• static			
• overloading			
▪ static variables			
▪ static final variables (a.k.a. constants)			
▪ null type			
▪ Inheritance			
• interface			
• abstract			
• extends			
• implements			
• super class			
○ calling the constructor			
○ calling a method			
○ using this			

Topics:	Level of Understanding		
	Huhh?	Got It	Can Teach It
• Standard Java Libraries/Classes			
○ System.out			
▪ print			
▪ println			
○ String class			
▪ Methods			
• compareTo			
• equals			
• length			
• substring			
• indexOf(String s)			
▪ Escape sequences in literal strings			
• \n (a.k.a. new-line)			
• \\ (a.k.a. "so how do I make a \ then?")			
• \" (a.k.a. the quote mark)			
▪ Interface			
• comparable<T>			
○ Integer class			
▪ Methods			
• Integer(int x) - Constructor			
• intValue			
• toString			
• equals			
• compareTo			
▪ Constants			
• .MIN_VALUE, .MAX_VALUE			
▪ Interface			
• comparable<T>			
○ Double class			
▪ Methods			
• Double(double x) - Constructor			
• doubleValue			
• toString			
• equals			
• compareTo			
▪ Interface			
• comparable<T>			
○ Math			
▪ abs(int x) (a.k.a. absolute value of integer)			
▪ abs(double x) (a.k.a. absolute value / real number)			
▪ pow(double base, double exp) (a.k.a. $base^{exp}$ )			
▪ sqrt(double x) (a.k.a. $\sqrt{x}$ )			
▪ random() (a.k.a. random number between 0-1)			
○ List<E> Interface			

Topics:	Level of Understanding		
	Hunh?	Got It	Can Teach It
○ ArrayList<E>			
▪ Constructor			
▪ add			
▪ get			
▪ set			
▪ remove			
▪ size			
▪ equals			
• Exceptions			
○ NullPointerException (a.k.a. no return value error)			
○ ArrayIndexOutOfBoundsException (a.k.a. array index error)			
○ ArithmeticException (a.k.a. divide by 0 error)			
○ IllegalArgumentException (a.k.a. wrong type parameter error)			
○ ClassCastException (a.k.a. Not a descendent class error)			
• Recursion			
• Preconditions			
• Post-conditions			