**Video Name:** OO Basics

**Topics:**
- Objects and Classes
- Constructors, getters and setters
- Constants
- Initializing variables

**Java Class(es):** MyPoint

**Vocabulary:**
- **instance variable.** Each object has its own copy
  - instance variables should generally be private
- **constructor (ctor)**
  - methods used to initialize objects
  - Eclipse can generate
  - super calls parent ctor (not needed if no parent variables, but doesn’t hurt)
  - name of ctor matches name of class
- **accessors**
  - setters – generally take parameter, assign to instance variable
  - getters – generally just return object, OR copy of object

**Java Class(es):** Book

**Constant:** public static final int MAX_HIGHLIGHTS = 100;
- OK to be public (can’t change value anyway, no need to protect)
- static – only one copy. Since can’t change, no need for each object to have its own copy.
- final – this is what makes it constant
- name in all caps – convention for all constants
- = value – sets the value of the constant

**Initializing variables**
- Java automatically initializes instance variables (e.g., int set to 0, object set to null, etc)
- Programmer must initialize local variables before use (Java enforces)

**Semantics (software engineering!)**
- pick method names that match desired functionality
- rather than just using setters, use meaningful names like `turnThePage`
- use descriptive function names! Slightly longer, more meaningful names are preferable.
Common error (software engineering)

- off by one (e.g., can’t get to last page)

Print format

- use tab escape sequence: `System.out.println("\t" + highlights[i]);`