Module 1 - Lecture 13

Managing Inheritance

Review

- What is polymorphism?
- What are two ways to achieve polymorphism?
- What is an interface?

Abstract

- Abstract classes cannot be instantiated, but they can provide logic and structure to their subclasses.
- Abstract methods are methods without logic and must be implemented by their subclasses.
- If a class includes an abstract method, it must be marked as abstract.
- If a class derives from an abstract class, it must override all abstract methods from its parent or it must also be marked abstract.

Abstract vs Interface

- Abstract classes can have implemented methods. An interface cannot (kind of).
- Method definitions in an interface are considered abstract by default. In an abstract class abstract methods must be marked as abstract.
- A class can only inherit from one class, but can implement many interfaces.
- Inheriting from a superclass is akin to making a more specialized version of that class. Implementing an interface means that you will provide the required behavior(s). For example, there is a difference between being a Book and a TextBook vs being Sellable or Readable.

Final keyword

- Making variables final means that their value cannot change once it is set.

- Making methods final means that children cannot override what the parent has defined.
  - Prevents logic that is integral to the application from being overridden by a poorly behaving subclass.
  - A design decision that should have good reason for being.

- Making class final means that another class cannot inherit from it.
  - A design decision that should have good reason for being.

Access Modifiers

Modifier	Class	Package	Subclass	Global
Public	Yes	Yes	Yes	Yes
Protected	Yes	Yes	Yes	No
Default	Yes	Yes	No	No
Private	Yes	No	No	No

toString()

- Method inherited from java.lang.Object

- Can be overridden to define how to convert a given object into a String.
  - Override when you plan on printing an object to the screen or elsewhere.

Let's Code!

QUESTIONS?