**Pop Quiz 10/11**

**What is the benefit of separating your model code from your view code?**

You can add multiple user interfaces without needing to change your model

Encapsulation, separate business logic from customer logic

**What is the difference between specialization and generalization?**

Specialization - Base classes first, then create a specialized class out of those base classes

Generalization - Specialized classes first, then create a base class out of those specialized classes

**Draw a class diagram with a base class that implements subclass1 and subclass2.**



**What is realization?**

The implementation of an abstraction.

When a class implements an interface.



**What is the dependency inversion principle?**

You want concrete things like classes to depend on abstract things like interfaces

High level modules (Class/package/etc) should not depend on low level modules

Abstractions should not depend on details, details should depend on abstractions

**What is the main problem that the dependency inversion principle solves?**

It reduces tight coupling.

**Why would you want to use constructer injection?**

It’s when you want to create an object to work with only one specific dependency

You create the object, you put in the dependency, and then that module won't be changed by other code

**Why would you want to use method injection?**

Method injection is passing in dependencies as a method parameter

You would want to give the caller the power to satisfy that dependency, like the strategy pattern

**Between the three methods of dependency injection, what should you default to?**

The default should be constructer injection because of the principle of least knowledge, hide the implementation principles as much as possible, whatever is using those objects does not need to know which dependencies it’s using