**Where we are:**

|  |  |
| --- | --- |
| Intro to MVC | Midterm, Navigation |
| Controllers and Views | Forms and Tag Helpers |
| Unit Testing, Publishing to Azure | Annotation and validation |
| Dependency Injection, Repository Pattern and Models | Authentication and Authorization |
| Entity Framework and Scaffolding | More Authorization Final |

**Announcements**

**Review**

**Models**

Domain Model vs. Logical Model

Inheritance: is-a, triangle  
Example: A student is a person

Composition: has-a, black diamond  
Example: A student has an address

Aggregation: whole-part, open diamond  
A student is a part of a class

Domain model: in real life the student may have

Unit Testing

Dependency Injection and the Repository Pattern

* Dependency injection – control dependency at run-time.
* For testing, use a fake version of the database
* Use different database contexts depending on whether we’re doing testing or not? No, the database context is too complex to simulate.
* Create a repository class that is an abstraction layer over the database context. It is typically put in a project folder named DAL (Data Access Layer).

Null Conditional Operator

References

* [ASP.NET MVC Controller Dependency Injection for Beginners, by S. M. Ahasan Habib, 31 Dec 2013](http://www.codeproject.com/Articles/560798/ASP-NET-MVC-controller-dependency-injection-for-be)