Java Notes

# General Notes

* JDBC is a very low-level API (think ADO.NET)
* [JSP vs. Servlets vs. JSF](http://stackoverflow.com/questions/2095397/what-is-the-difference-between-jsf-servlet-and-jsp)
* **final** equates to .NET’s **sealed**
* Use **generic** interfaces (similar to C#) to guarantee type-safety
* **JavaFX** - framework to create rich client applications

# Primitives/Primitive Wrapper Classes

* **Primitive** types are true primitives, unlike C#’s value types. They are not classes.
* Primitives use less memory and should be preferred unless functionality from the corresponding type is needed
* Primitive types are stored by value
* int/Integer
* Integer a = Integer.valueOf(100); // primitive to wrapper
* int b = a.intValue();

Wrapper Class Equality

* Integer a = 1 + 1;  
  Integer b = 2;  
  a == b is false, but a.equals(b) is true

# Java EE

* Extends Java SE (core Java)
* Enterprise software, large scale
* Convention over configuration, e.g. annotations for entities to make them database-ready

# JRE vs. JDK

## JRE

* Required to run Java apps
* End users normally require only the JRE

## JDK

* Provides tools required to create Java apps
* Developers require the JDK
* JDK installation includes JRE

# Initialization

* **Constructor-chaining** uses this syntax: this(argument)
* **Initialization blocks** are executed for each new instance of the class
* **Static initialization blocks** are executed once

## Initialization Order

1. Field initializers
2. Initialization blocks
3. Constructors