Checklist review today:

Design Pattern

* Singleton
  + EagerLoad -

public class EagerLoad{

private static final EagerLoad instance = new EagerLoad();

private EagerLoad(){

if (instance != null)

throw new RuntimeException (“Can’t create!”);

}

public static EagerLoad getinstance(){

return instance;

}

private void dosomething(){System.out.println(“1234”);}

}

* + LazyLoad -

public class LazyLoad{

private static volatile LazyLoad instance = null;

private LazyLoad(){}

public static LazyLoad getinstance(){

if(instance == null){

synchronized (LazyLoad.colass){

if(instance == null){

instance = new LazyLoad();

}

}

}

return instance;

}

}

* + Factory
    - Creational Design pattern
    - minimize coupling
    - hide detail of how we create object (to API caller)
  + builder
    - simplify detail of how we create object (to API caller)
    - inner builder class
  + Proxy
    - in place of
    - Static Proxy
    - Dynamic Proxy

Learned Plan tomorrow:

* + Review OSI & TCP/IP
  + Review HTTP
    - request
    - response
    - status code
    - method