1. The abstract common services tactic is intended to reduce coupling, but it also might reduce cohesion. Why?  
   抽象公共服务策略是指当存在两个不同的功能模块提供相似的服务时，使用一个更为一般化的形式抽象出类型与参数，以减少修改的位置个数和修改次数。这样模块间的进一步抽象很显然降低了模块间的耦合度，因为进一步封装让模块间的接口更少，相互之间的影响更小，独立性更强。但同时，这种策略模式还可能减少内聚性，因为高度抽象出来的参数一般是和与之相似模块的功能重叠和逻辑一致的部分，而不是完全相同的功能和逻辑，这与增加内聚性的方法——不同的功能为两个不同的模块，恰好相反。
2. Every system has real-time performance constraints. Discuss. Or provide a counterexample.  
   每个系统都有一个实时的性能限制，它是由系统本身的构造决定，不能做到完全消除。理由如下：  
   1. 计算机的资源有限，资源消耗会产生性能约束。资源包括cpu、数据存储、通信网络带宽和内存，也可以包括设计中的特定系统定义的实体。对于资源的消耗上，就会产生时间上的约束。  
   2. 闭锁时间。计算的运行可能会由于资源争用、资源不可用、计算依赖而被阻止，在此基础上就会产生响应时间。  
   3. 处理时间。任何系统，对一个服务都会进行处理，而进行相应处理是需要时间的。
3. Web-based systems often use proxy servers, which are the first element of the system to receive a request from a client(such as your browser)。Proxy servers are able to serve up often-requested web pages, such as a company’s home page, without bothering the real application server that carry out transactions. There may be many proxy servers, and they are often located geographically close to large user communities, to decrease response time for routine request. What tactics do you see at work here?  
   有以下策略被使用：  
   1. 增加资源的策略（increase resources）。代理服务器的使用，需要额外的处理器、内存以及更快的网络，以减少响应时间。  
   2. 维护多个计算副本和维护多个数据副本的策略（maintain multiple copies of computations & maintain multiple copies of data）。代理服务器上，存放着存放着一些常用的数据或服务，以减少主服务器的压力，同时减少响应时间。  
   3. 减少日常开销的策略（reduce overhead）。将服务器安排在距用户近的位置，减少了日常访问时间。  
   4. 调度资源策略（schedule resources）。将经常访问网页置于代理服务器中，提高计算效率，类似于cache的原理，提高了资源调度效率。  
   5. 增加资源有效性（increase resource efficiency）。采用类似cache的机制，使资源能更加地被合理利用，极大增加资源有效性。