**A1. Briefly discuss key challenges facing today’s information system development**

Changing computer environment; changing business requirements; scale-up in multiple dimensions.

**A2. We claimed in class that OO approaches and software design patterns are most useful when dealing with ``real-world'' software development problems. Briefly discuss at least three dimensions of this ``real-word''-ness.**

**A3. What are the four pillars of Object Orientation? Briefly explain each element in 1-2 sentences.**

* Abstraction: it’s a function way to deal with complexity. It focuses on the essential characteristics of the object from the perspective of views.
* Encapsulation: it’s the process of compartmentalizing the elements of an abstraction. It focuses on the implementation that gives rise to the behavior. Encapsulation serves to separate the contractual interface of an abstraction and its implementation.
* Modularity: it is the property of a system that has been decomposed into a set of cohesive and loosely coupled modules
* Hierarchy: it is a ranking or ordering of abstraction. The two most important hierarchies in a complex system are its class structure and its object structure.

**A4. For a system, which one is a better structure: a) low cohesion & tight coupling, b) high cohesion & loose coupling? Give a system example (for your better structure choice) and briefly explain cohesion and coupling characteristics of this system.**

The better structure for a system is high cohesion and loose coupling.

**B.1. Why is abstraction so important for software design? Why is abstraction so important for software integration?**

* Abstraction could emphasize details that are significant to the reader or user. It focuses on the outside view of an object and separates an object’s essential behavior from its implementation.
* Abstraction could help every object collaborates with other objects to achieve some behavior. The system design about how the objects cooperate with one another defines the boundaries of each abstraction and the responsibilities of each object.