5.1. [20 points] Are the following statements true or false? Explain why you chose true or false.
a) The Information Hiding Principle refers to hiding design decisions in a module.

False. It put key design decisions in separate modules.

b) A Module Guide describes the implementation of each module.

True. The hidden implementation is the module's secret and for a module guide each module described by its secret.

c) A module with a secret cannot be changed.

False. The information hiding for each module is built for change. Though, change of on module cannot affect other modules.

d) A Module Interface Specification specifies the services provided and the services needed by modules.

True. Based on the description from slides.

e) With the XP focus on the simplest thing that could possibly work and on refactoring to clean up the design as new code is added, there is no need for architecture.

False. It still needs a design so architecture would help the development.

f) A system that obeys the Information Hiding principle is secure.

True. Since it will be impossible to change a module in that case.

g) If module A uses module B, then A and B must have an ancestor-descendant relation in the module hierarchy.

False. Since module B might use A then they will have a loop relation.

h) If module A uses module B, then B must be present and satisfy its specification for A to satisfy its specification.

True. Based on the definition of the uses relation.

i)A Development View of an architecture specifies the internal structure of components.

False. It specifies modules in the static source instead.

j) Conway's "law" implies that the architecture of a system reflects the social structure of the producing organization.

True. The interface structure of the system necessarily reflects the social structure of the organization that produced it.

5.2. [20 [points] Coupling is the degree to which modules are inter-related. Forms of coupling include:

a) ***Message***: pass messages through their interfaces.

b) ***Subclass***: inherit methods and data from a superclass.

c) ***Global***: two or more modules share the same global data.

d) ***Content***: one module relies on the implementation of another.

For each of the above cases, suppose modules ***A*** and ***B*** have that kind of coupling. How would you refactor ***A*** and ***B*** into modules M_1, M_2, $\cdots$ that comply with Information Hiding and provide the same services as ***A*** and ***B***. That is, for each public function ***A.f()*** or ***B.f()*** in the interfaces of ***A*** and ***B***, there is an equivalent function $M_i.f()$, for some refactored module M_i.

Since some of the modules share the same data, so I would say, to be careful, modify both the public and private(secret) functions but do not change the function that is built for the super class.

Also, be careful with the messages passing.