**Module 3 Review**

**LATEST SUBMISSION GRADE**

75%

1.Question 1

Which of these terms are used to describe coupling? **Choose the 3 correct answers.**

**1 / 1 point**



exposed



degree

**Correct**

Correct! Degree is how much two components are connected



flexibility

**Correct**

Correct! Flexibility is how easily a component can be used for another purpose.



frequency



ease

**Correct**

Correct! Ease is how easily a component can be switched for a different one.

2.Question 2

Which is the most desirable?

**1 / 1 point**



low cohesion, loose coupling



low cohesion, tight coupling



high cohesion, tight coupling



high cohesion, loose coupling

**Correct**

Correct!

3.Question 3

What are some keywords you might use for information hiding in Java? **Select the three correct answers.**

**0 / 1 point**



abstract



private

**Correct**

Correct! This will hide variables or methods from all other classes.



protected

**Correct**

Correct! This will hide information from all classes except those in the same package or those that extend.



[none]

You didn’t select all the correct answers

4.Question 4

What are the best ways to promote Conceptual Integrity in your software? **Choose the two correct answers.**

**1 / 1 point**



Regular code reviews

**Correct**

Correct! Regular code reviews will get everyone to think together about the best conventions to use in your software, and allow the team to correct missteps.



Delegating development of different components to different teams



Good commenting



Planning the architecture of the system

**Correct**

Correct! Planning ahead will allow your team to discuss issues relating to conceptual integrity ahead of time.

5.Question 5

**Information Hiding** is closely related to one of the core design principles of object-oriented design. Which one?

**1 / 1 point**



decomposition



abstraction



encapsulation



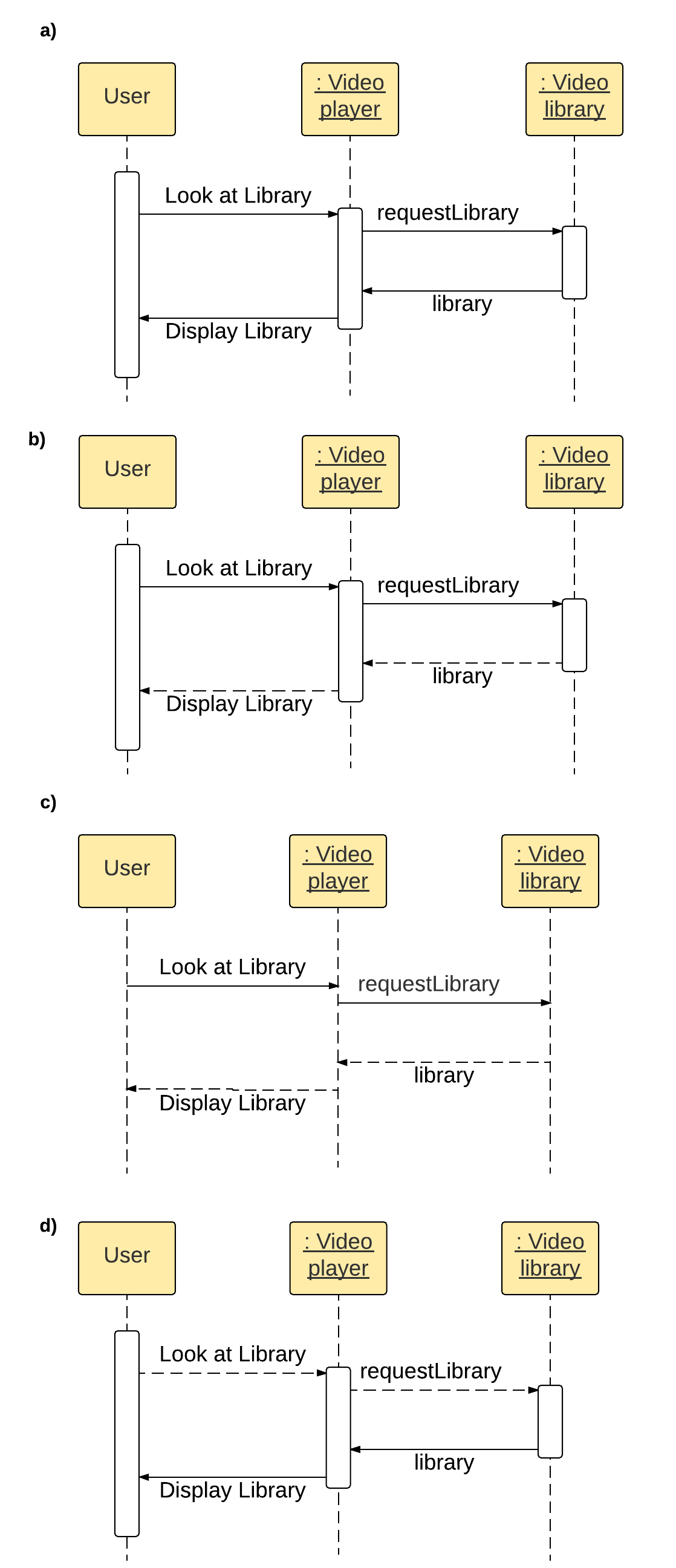
generalization

**Correct**

Correct! Information hiding involves hiding away those details that are not important to outside classes. This is closely related to encapsulation.

6.Question 6

Which of these sequence diagrams is correct?



**1 / 1 point**



**a)**



**b)**



**c)**



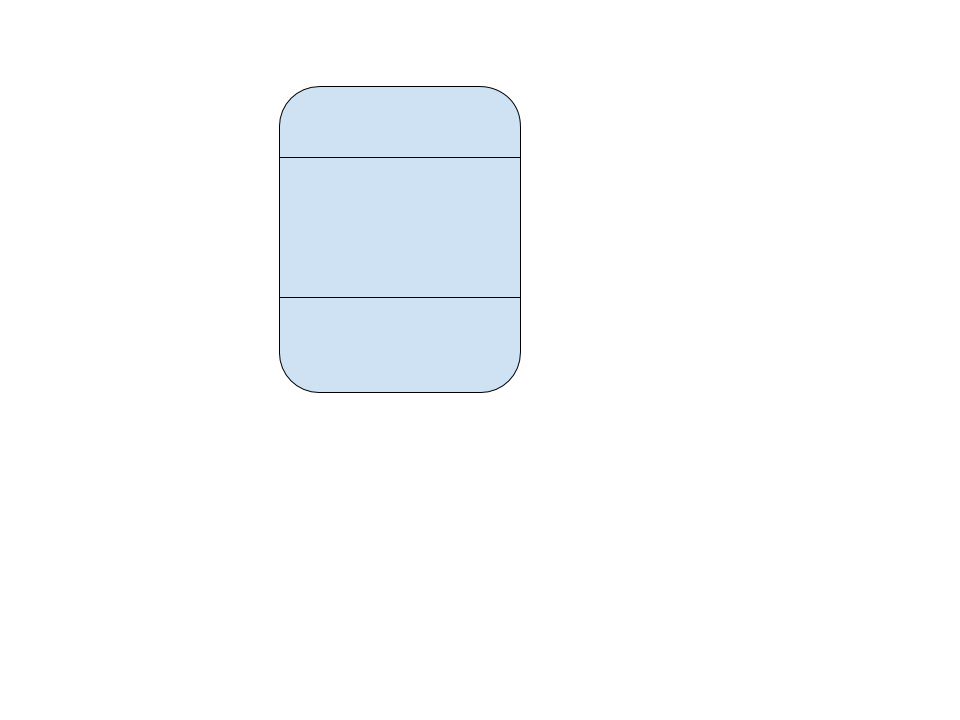
**d)**

**Correct**

Correct! This is a complete sequence diagram.

7.Question 7

What are elements of a state in a State diagram (see diagram)? **Choose the three correct answers.**



**1 / 1 point**



activities

**Correct**

Correct! The activities that are specific to this state are listed, sometimes including those that occur when entering or exiting the state.



responsibilities



state variables

**Correct**

Correct! State variables are manipulated depending on the state.



state name

**Correct**

Correct! The name of the state is at the top section of a state.



events

8.Question 8

When is **Model Checking** conducted?

**0 / 1 point**



During development



After development



During planning



After deployment

**Incorrect**

Incorrect. The goal of model checking is to catch errors before they make it "into the wild"

9.Question 9

What are the phases of Model Checking? **Choose the 3 correct answers.**

**1 / 1 point**



Modeling Phase.

**Correct**

Correct! First the team creates a model for testing the software in all of its different states.



Running Phase

**Correct**

Correct! The model checking software identifies counterexamples if there are any.



Analysis Phase

**Correct**

Correct! The counterexamples that were identified in the running phase are analysed to find the causes of the issues and the solution to each.



Model Simulation



Counterexample Phase

10.Question 10

During model checking, what is the name for a violation of the desired properties of the model?

**0 / 1 point**



Model Gap



Error



Counterexample



Redevelopment

**Incorrect**

Incorrect. This is not the term used by developers!

11.Question 11

When two processes cannot run because they are waiting on the same resource, it's called…

**1 / 1 point**



Transition lock



State lock



Deadlock



Mutual lock

**Correct**

Correct! This is called a deadlock.

12.Question 12

Choose the **three** examples of inheritance used **poorly**:

**1 / 1 point**



Inheritance is used to share behaviour without specializing

**Correct**

Correct! If inheritance is merely used to share behaviour and not much more, consider skipping it altogether and just using the superclass.



A method in the superclass is overwritten with different behaviour by a subclass.

**Correct**

Correct! This violates Liskov's Substitution Principle, which states that a superclass should be able to be substituted by a subclass without error.



The subclasses inherit methods from the superclass and have their own specific, related methods.



A subclass inherits methods from the superclass and adds extra, new, unrelated functionality

**Correct**

Correct! If your subclass inherits some behaviour and adds unrelated functionality, it is not very coherent. You should consider decomposing these responsibilities into different interfaces.