

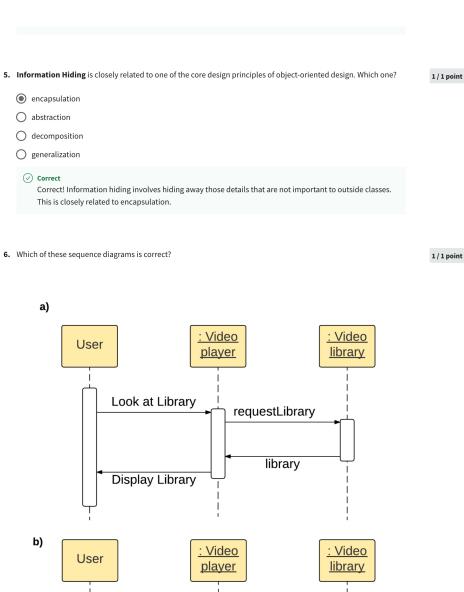
## Congratulations! You passed!

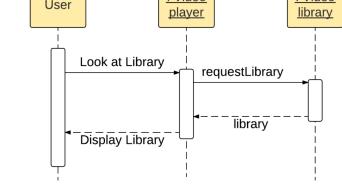
Grade Latest Submission received 100% Grade 100%

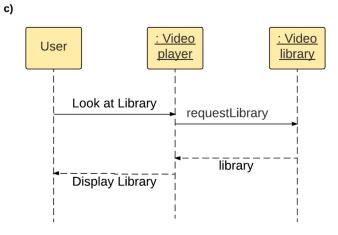
**To pass** 75% or higher

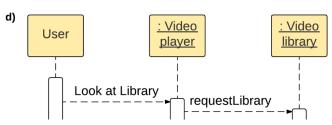
Go to next item

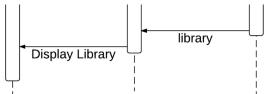
1.	Which of these terms are used to describe coupling? <b>Choose the 3 correct answers.</b>	1/1 point	
	<ul> <li>Correct         Correct! Ease is how easily a component can be switched for a different one.     </li> </ul>		
	☐ frequency  ✓ flexibility		
	<ul> <li>✓ Correct         Correct! Flexibility is how easily a component can be used for another purpose.    </li> </ul>		
	<ul><li>□ exposed</li><li>✓</li></ul>		
	degree		
	✓ Correct     Correct! Degree is how much two components are connected		
2.	Which is the most desirable?	1/1 point	
	high cohesion, tight coupling  low cohesion, loose coupling		
	high cohesion, loose coupling		
	O low cohesion, tight coupling		
3.	What are some keywords you might use for information hiding in Java? <b>Select the three correct answers.</b>	1/1 point	
	<b>☑</b> private		
	<b>☑</b> protected		
	✓ Correct Correct! This will hide information from all classes except those in the same package or those that extend.		
	☑ [none]		
	✓ Correct  Correct! Having no keyword will make the variable or method default to only being accessible by the class and its subclasses.		
	abstract		
4.	What are the best ways to promote Conceptual Integrity in your software? <b>Choose the two correct answers.</b>	1/1 point	
	Delegating development of different components to different teams		
	<b>☑</b> 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.		
	Good commenting		
	✓ Planning the architecture of the system		
	<ul> <li>Correct         Correct! Planning ahead will allow your team to discuss issues relating to conceptual integrity ahead of time.     </li> </ul>		











7.

( a)	
○ c)	
○ d)	
○ Correct     Correct! This is a complete sequence diagram.	
What are elements of a state in a State diagram (see diagram)? <b>Choose the three correct answers.</b>	1/1 point
□ responsibilities	
✓ state name	
<ul><li>■ events</li><li>✓ activities</li></ul>	
Correct Correct! The activities that are specific to this state are listed, sometimes including those that occur when entering or exiting the state.	
<b>▼</b> state variables	
<ul> <li>Correct</li> <li>Correct! State variables are manipulated depending on the state.</li> </ul>	
When is <b>Model Checking</b> conducted?	1/1 point
After development	
During planning  During development	
O During development  After deployment	
<ul> <li>Correct         Correct! Model Checking is done after the bulk of the development is finished.     </li> </ul>	

Wha	it are the phases of Model Checking? <b>Choose the 3 correct answers.</b>
<b>~</b>	Modeling Phase.
0	Correct! First the team creates a model for testing the software in all of its different states.
<b>~</b>	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
	Running Phase
(~	Correct! The model checking software identifies counterexamples if there are any.
	Counterexample Phase
Duri	ng model checking, what is the name for a violation of the desired properties of the model?
0	Model Gap
0	Redevelopment
•	Counterexample
$\circ$	Error
0	Correct This is called a countergrammed
	correct! This is called a counterexample.  In two processes cannot run because they are waiting on the same resource, it's called
. Whe	Correct! This is called a counterexample.
Whe	Correct! This is called a counterexample.  In two processes cannot run because they are waiting on the same resource, it's called  Mutual lock  Transition lock
Whe	Correct! This is called a counterexample.  In two processes cannot run because they are waiting on the same resource, it's called  Mutual lock  Transition lock  Deadlock
Whe	Correct! This is called a counterexample.  In two processes cannot run because they are waiting on the same resource, it's called  Mutual lock  Transition lock  Deadlock  State lock  Correct
Whee	Correct! This is called a counterexample.  In two processes cannot run because they are waiting on the same resource, it's called  Mutual lock  Transition lock  Deadlock  State lock  Correct
Whee	Correct! This is called a counterexample.  In two processes cannot run because they are waiting on the same resource, it's called  Mutual lock  Transition lock  Deadlock  State lock  Correct  Correct! This is called a deadlock.
Whee O	Correct! This is called a counterexample.  In two processes cannot run because they are waiting on the same resource, it's called  Mutual lock  Transition lock  Deadlock  State lock  Correct  Correct! This is called a deadlock.  ose the three examples of inheritance used poorly:
Whee O	Correct! This is called a counterexample.  In two processes cannot run because they are waiting on the same resource, it's called  Mutual lock  Transition lock  Deadlock  State lock  Correct  Correct! This is called a deadlock.  ose the three examples of inheritance used poorly:  The subclasses inherit methods from the superclass and have their own specific, related methods.
Whee O	Correct! This is called a counterexample.  In two processes cannot run because they are waiting on the same resource, it's called  Mutual lock  Transition lock  Deadlock  State lock  Correct  Correct! This is called a deadlock.  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.
Whee Cho	Correct! This is called a counterexample.  In two processes cannot run because they are waiting on the same resource, it's called  Mutual lock  Transition lock  Deadlock  State lock  Correct  Correct! This is called a deadlock.  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.
Whee Cho	Correct! This is called a counterexample.  In two processes cannot run because they are waiting on the same resource, it's called  Mutual lock Transition lock Deadlock State lock  Correct Correct! This is called a deadlock.  Ose the three examples of inheritance used poorly:  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.  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