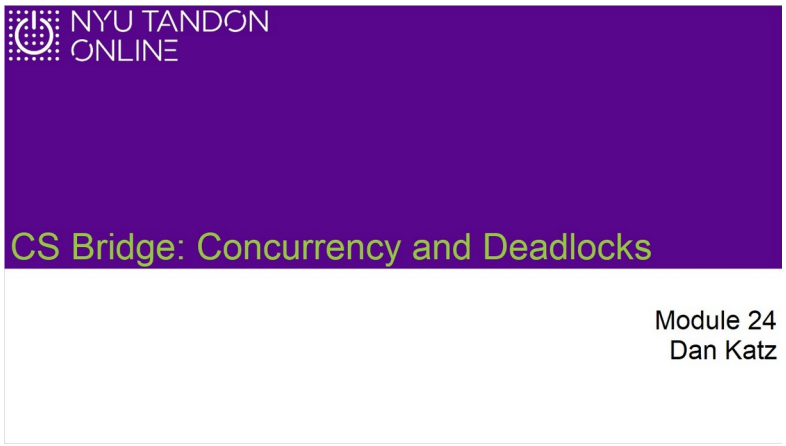
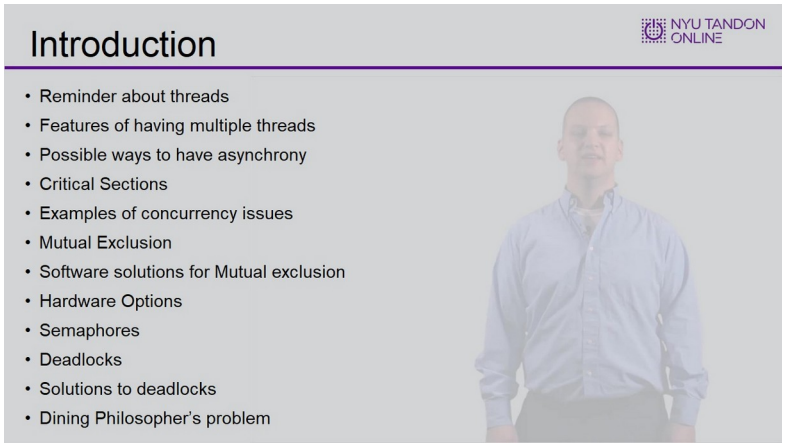
CS Bridge Module 24 Concurrency and Deadlocks

1. Concurrency and Deadlocks

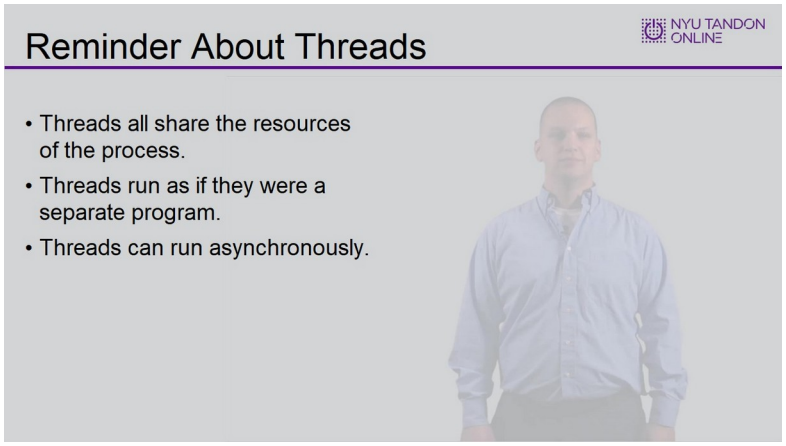
1.1 CS Bridge: Concurrency and Deadlocks



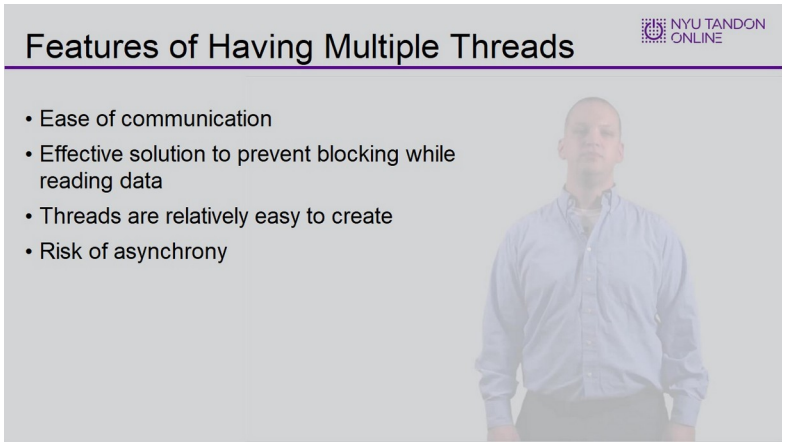
1.2 Concurrency and Deadlocks - Intro



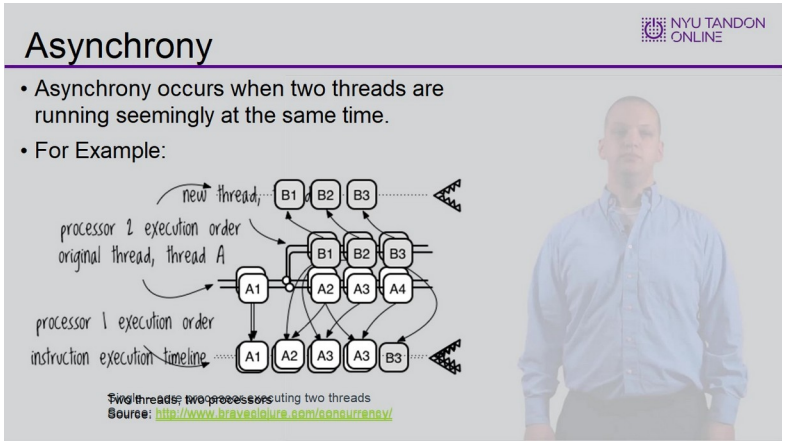
1.3 Reminder about threads



1.4 Features of having multiple threads



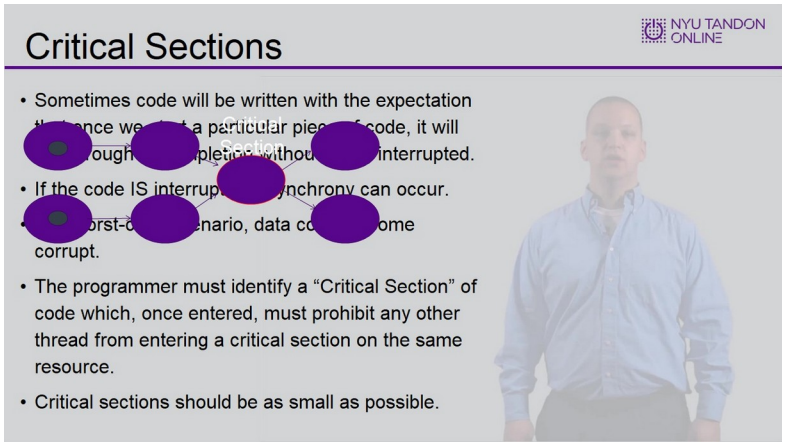
1.5 Asynchrony



Notes:

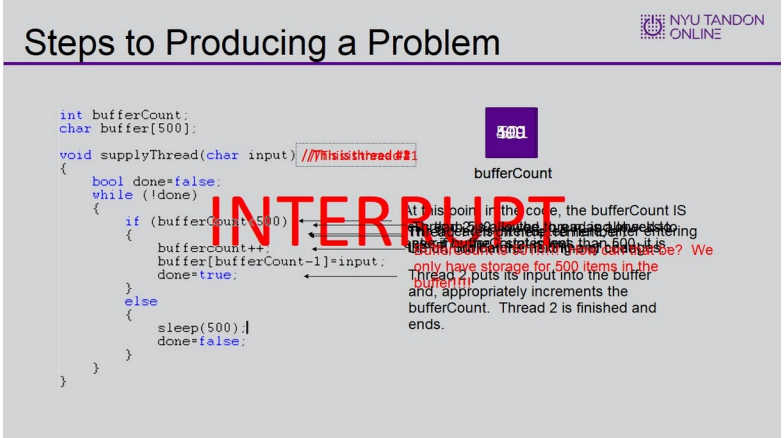
Needed Animations: 1) Two threads (wavy lines), different colors, in a box labelled “READY.” One thread moves to a box labelled “CPU/RUNNING.” Stays in the CPU box for a short time, then a red light labelled “INTERRUPT” appears next to CPU box. Thread moves from CPU box to READY box, short delay, INTERRUPT light goes out, then the other thread is moved from ready box to CPU box.

1.6 Critical Sections

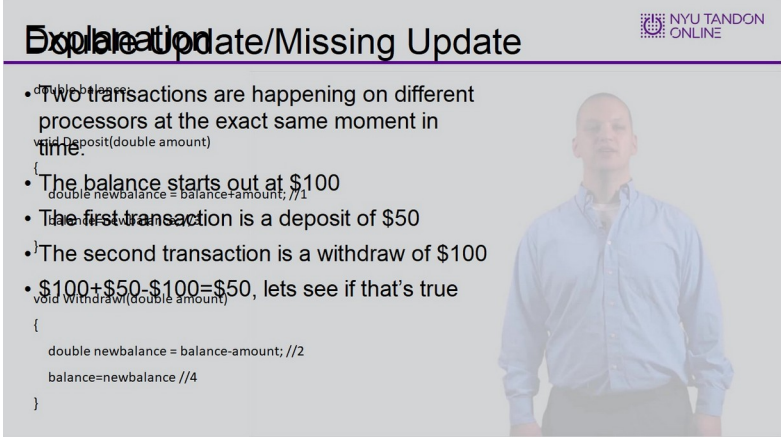


1.7 Supplier/Demander explanation

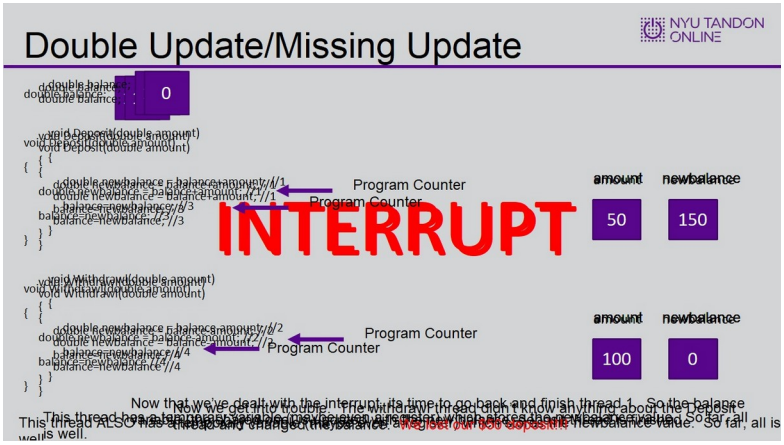
1.8 Steps to producing a problem -1



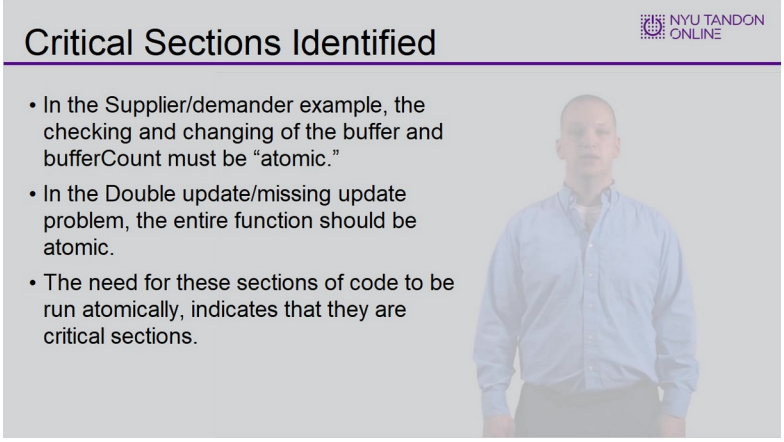
1.9 Double Update/Missing Update



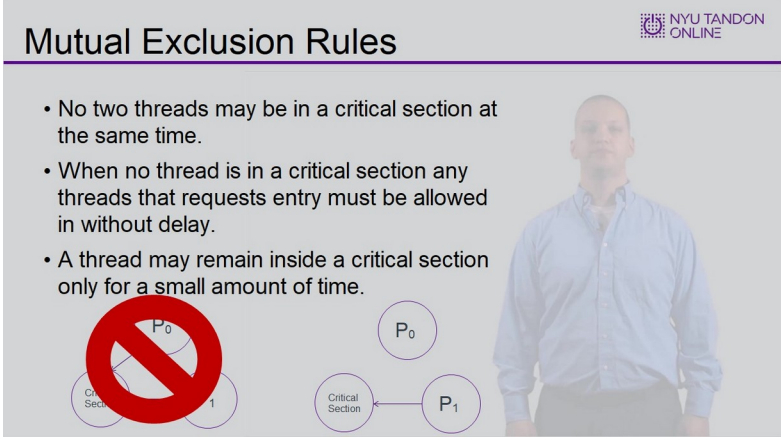
1.10 Double Update/Missing Update -1



1.11 Critical sections identified

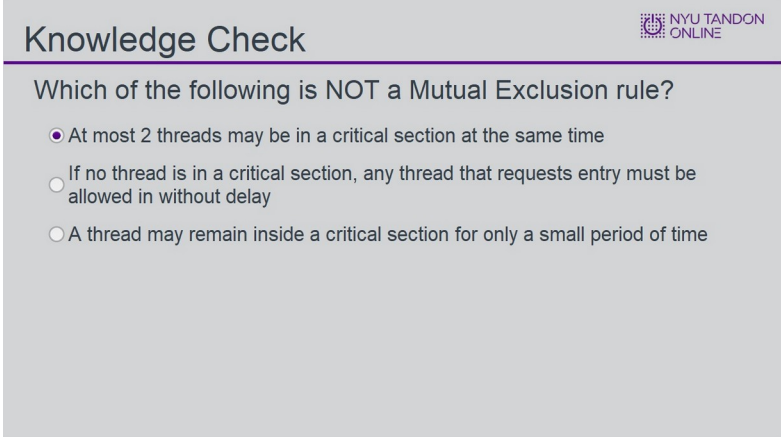


1.12 Mutual Exclusion rules



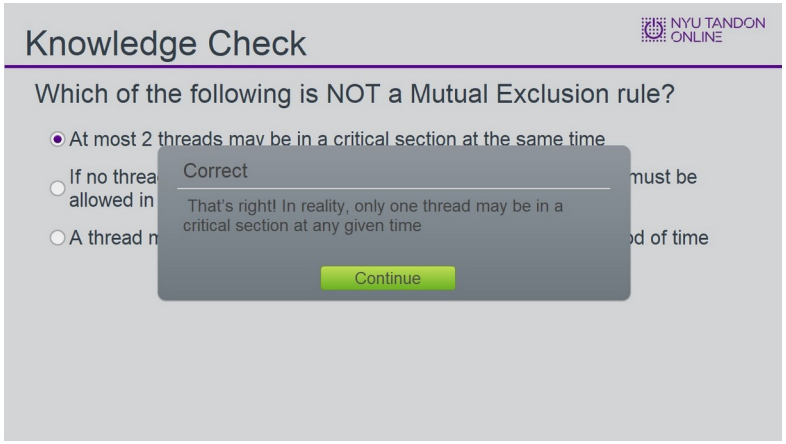
1.13 Knowledge Check

(Multiple Choice, 10 points, 4 attempts permitted)

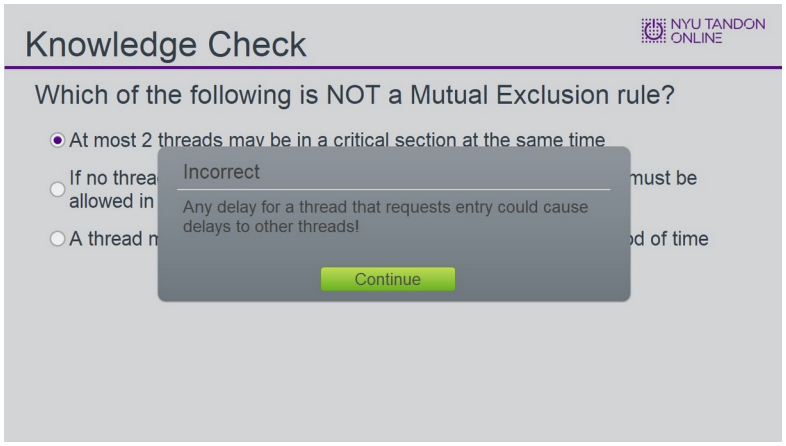


Correct	Choice	Feedback
X	At most 2 threads may be in a critical section at the same time	That's right! In reality, only one thread may be in a critical section at any given time
	If no thread is in a critical section, any thread that requests entry must be allowed in without delay	Any delay for a thread that requests entry could cause delays to other threads!
	A thread may remain inside a critical section for only a small period of time	A thread that stays inside the critical section for too long is depriving other threads of resources!

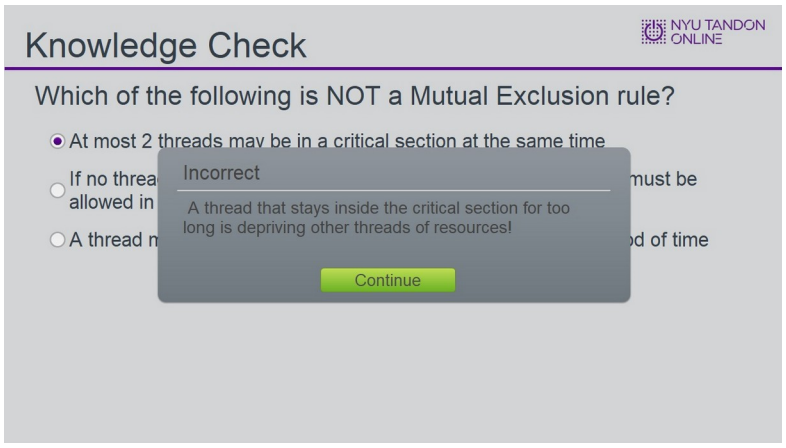
Incorrect (Slide Layer)



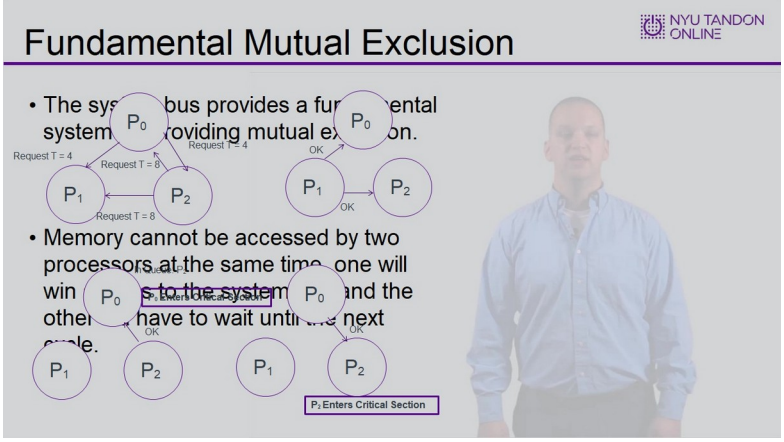
Incorrect (Slide Layer)



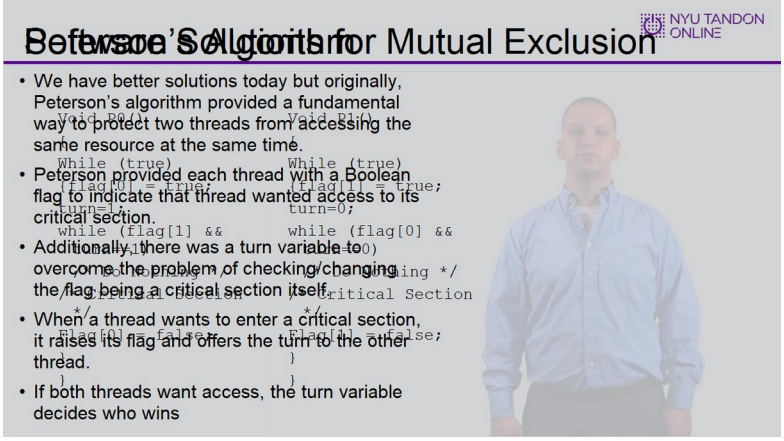
Incorrect (Slide Layer)



1.14 Fundamental mutual exclusion

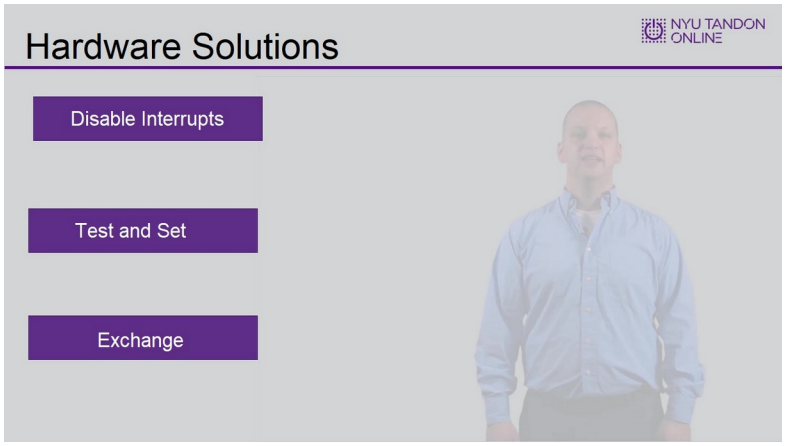


1.15 Software Solutions for Mutual Exclusion

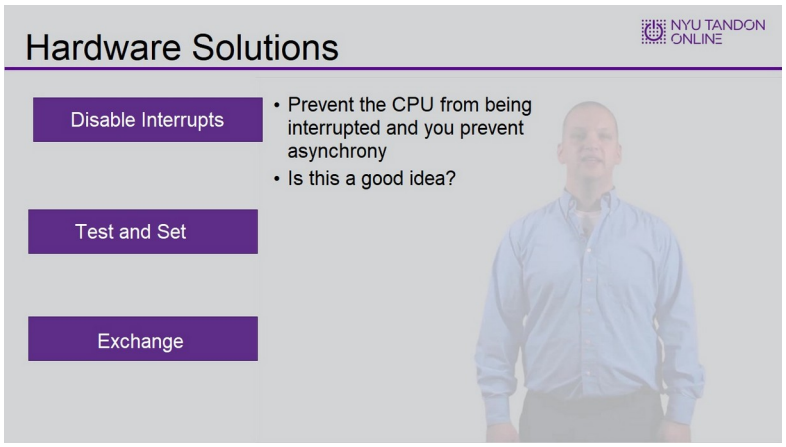


Notes:

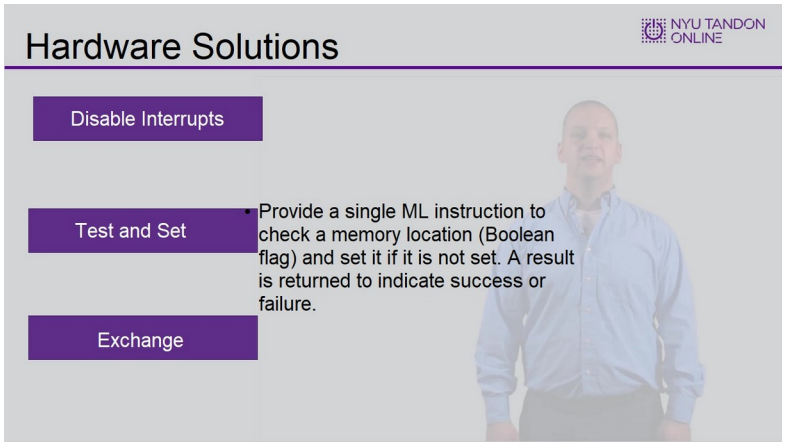
1.16 Hardware Solutions



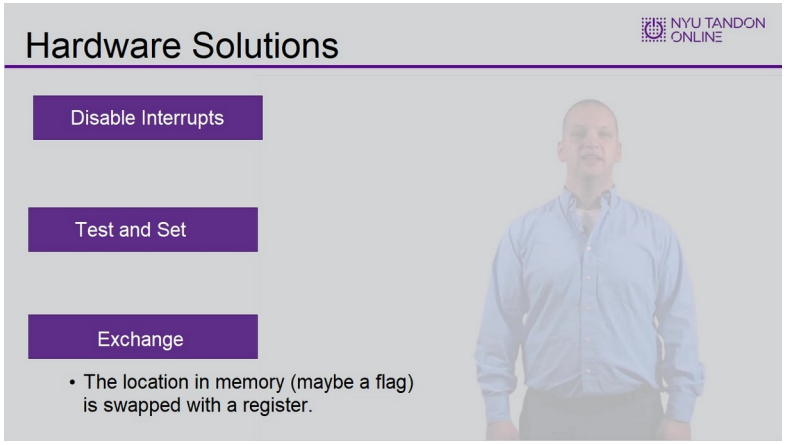
Disable Interrupts (Slide Layer)



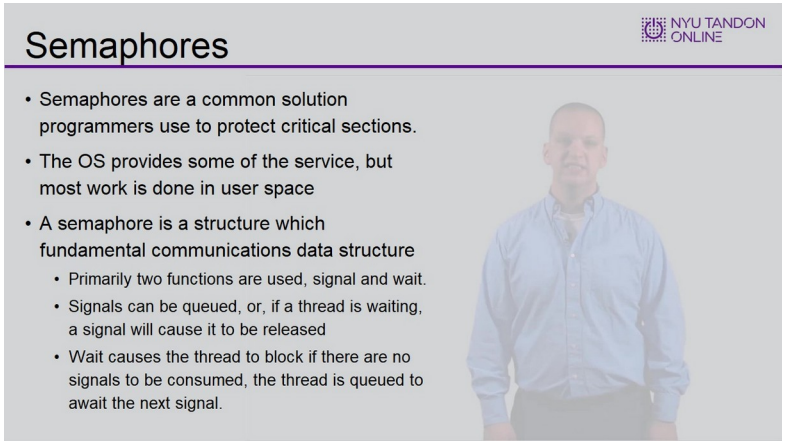
Teest and Set (Slide Layer)



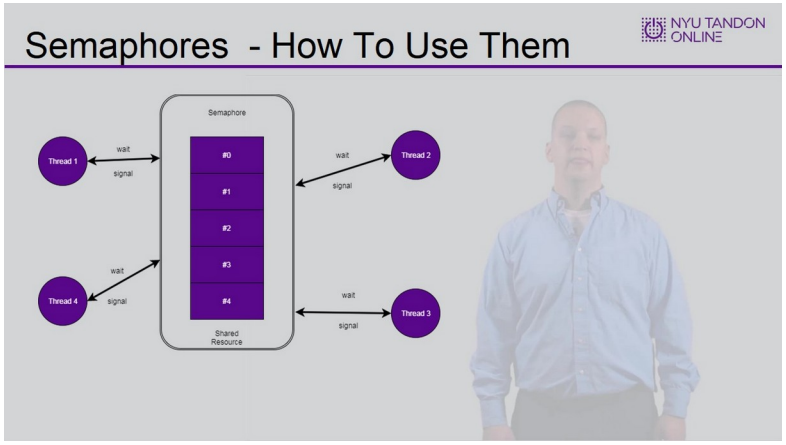
Exchange (Slide Layer)



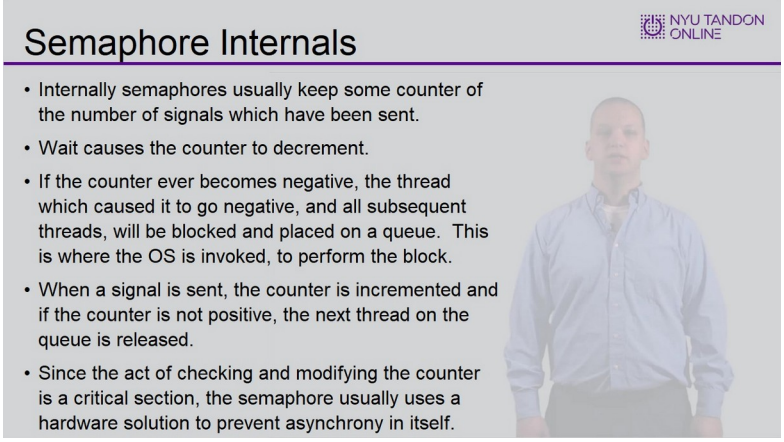
1.17 Semaphores



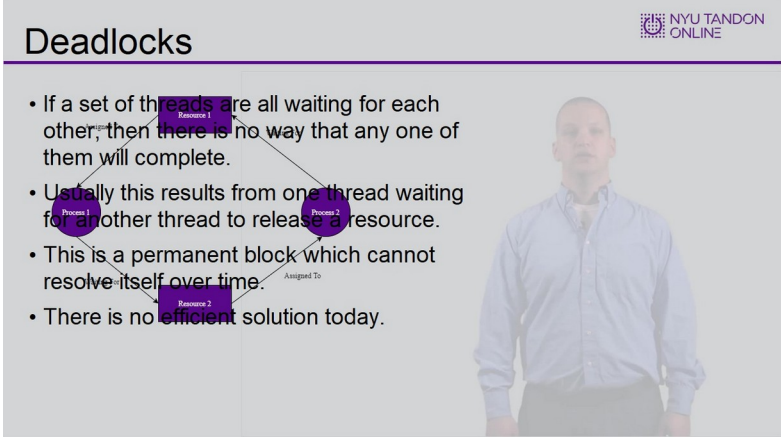
1.18 Semaphores - How to use them



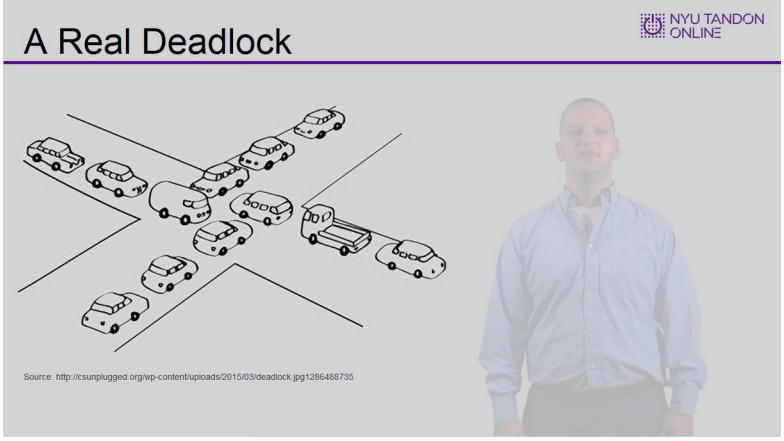
1.19 Semaphore internals



1.20 Deadlocks



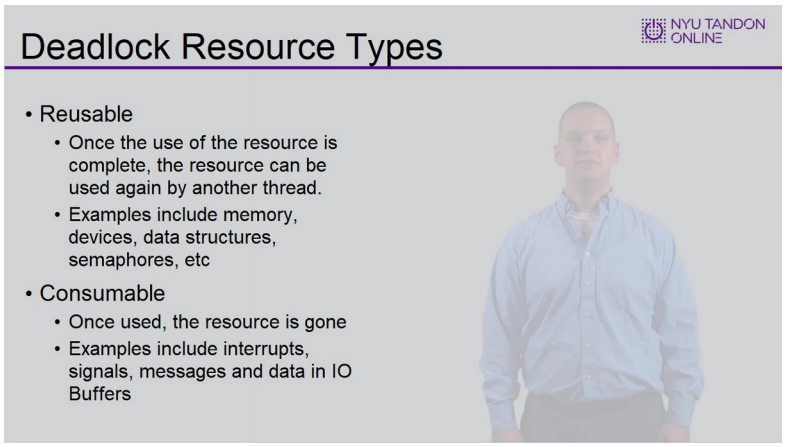
1.21 A real deadlock



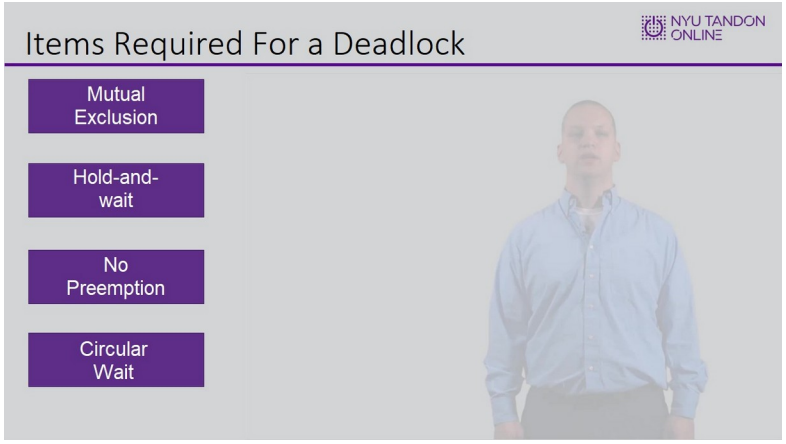
Notes:

Source: http://csunplugged.org/wp-content/uploads/2015/03/deadlock.jpg1286488735

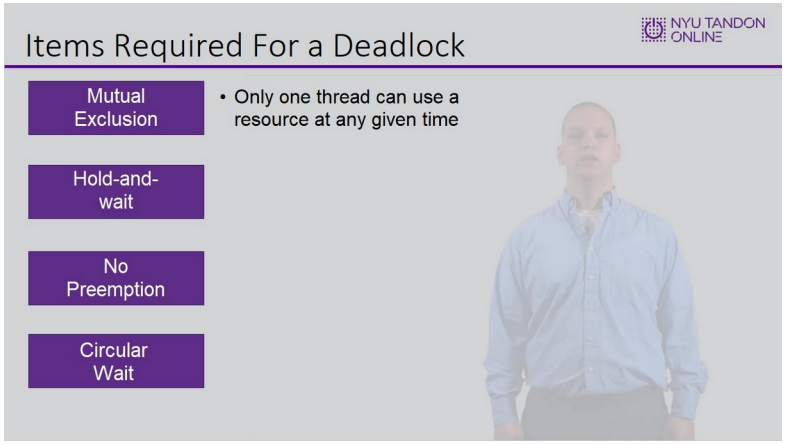
1.22 Deadlock resource types



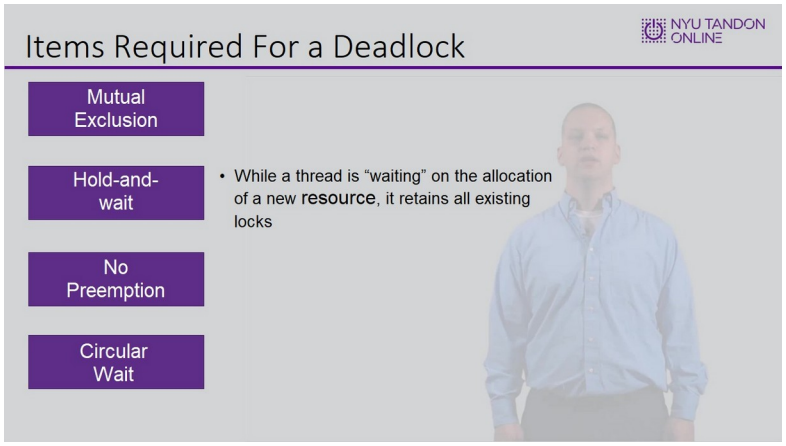
1.23 Items required for a deadlock



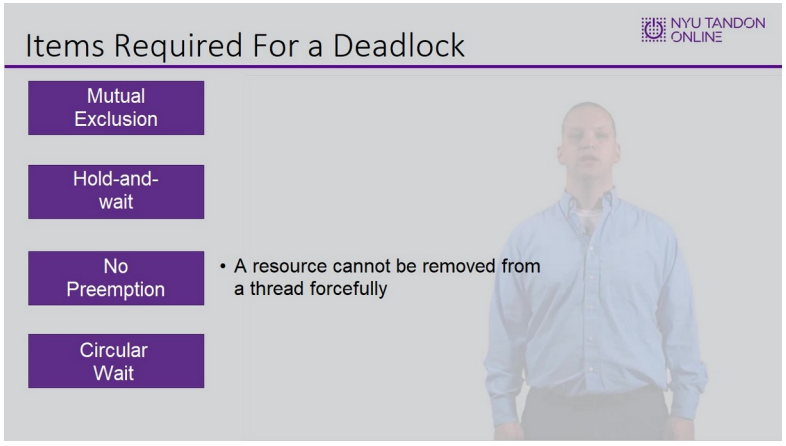
Mututal Exclusion (Slide Layer)



Hold And Wait (Slide Layer)

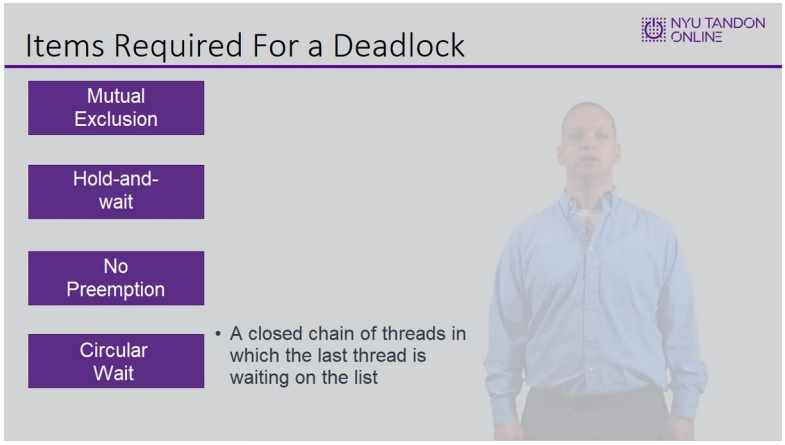


No Preemption (Slide Layer)



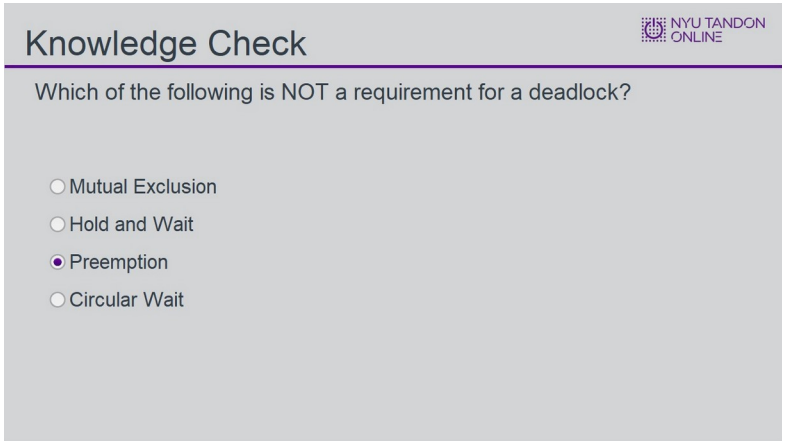
Published by Articulate® Storyline www.articulate.com

Circular Wait (Slide Layer)



1.24 Knowledge Check

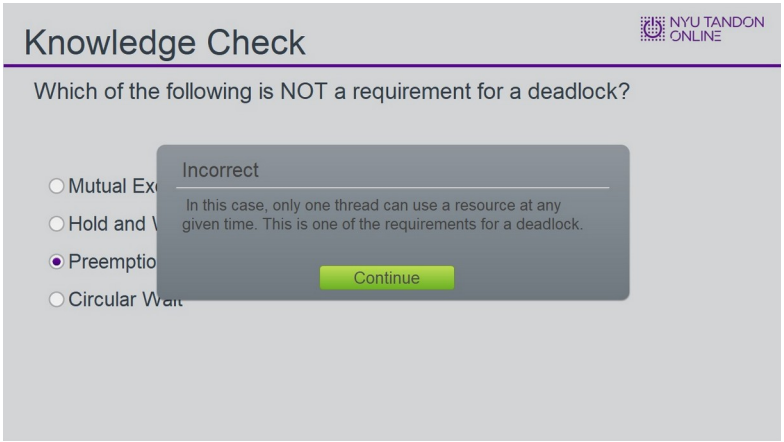
(Multiple Choice, 10 points, 4 attempts permitted)



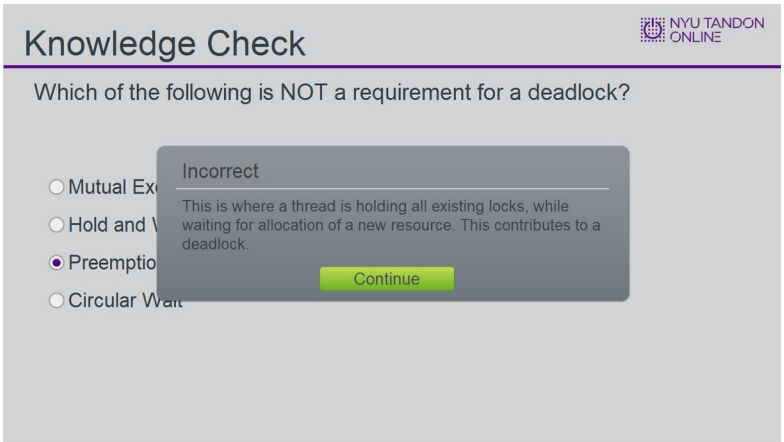
Correct	Choice	Feedback
	Mutual Exclusion	In this case, only one thread can use a resource at any given time. This is one of the requirements for a deadlock.
	Hold and Wait	This is where a thread is holding all existing locks, while waiting for allocation of a new resource. This

		contributes to a deadlock.
X	Preemption	Correct!
	Circular Wait	This is where in a closed chain of threads, the last thread is waiting for the first thread to finish. This contributes to a deadlock.

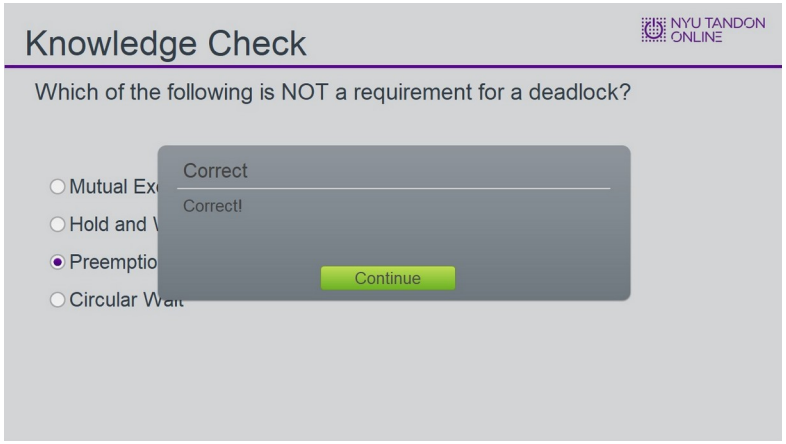
Incorrect (Slide Layer)



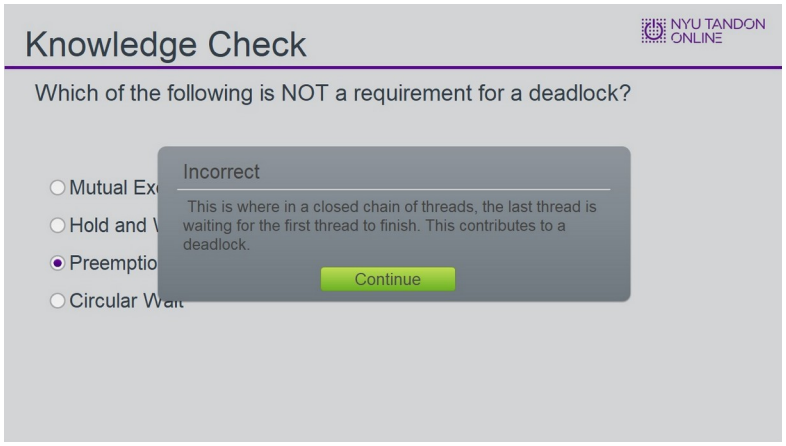
Incorrect (Slide Layer)



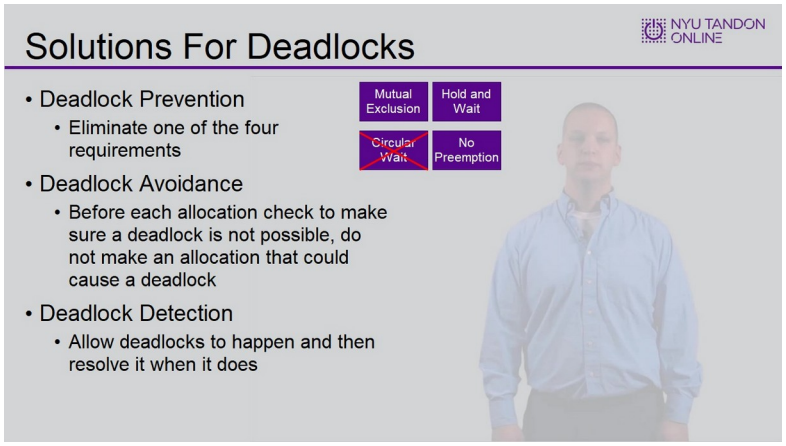
Correct (Slide Layer)



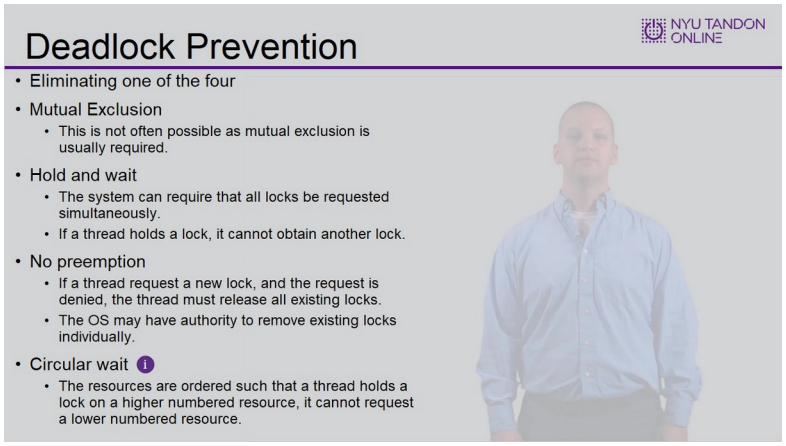
Incorrect (Slide Layer)



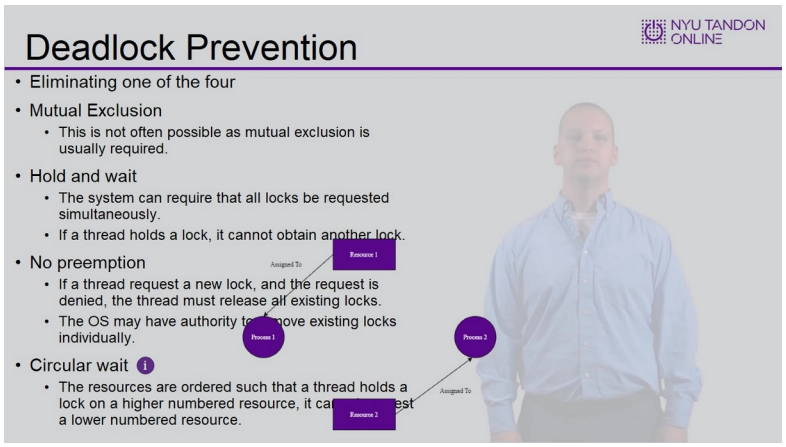
1.25 Solutions for deadlocks



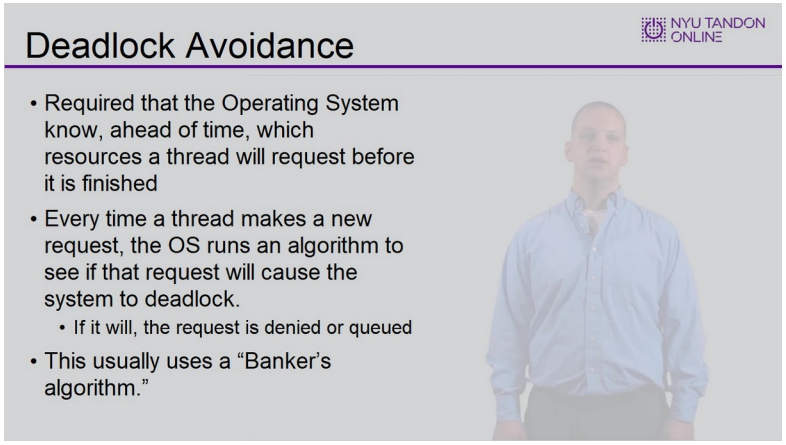
1.26 Deadlock Prevention



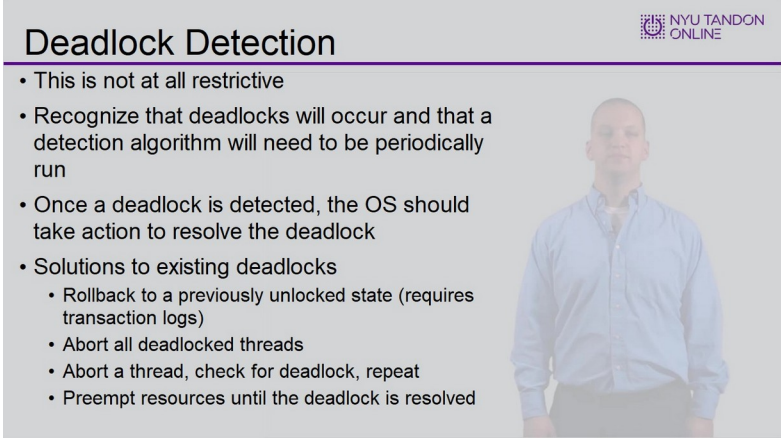
Untitled Layer 1 (Slide Layer)



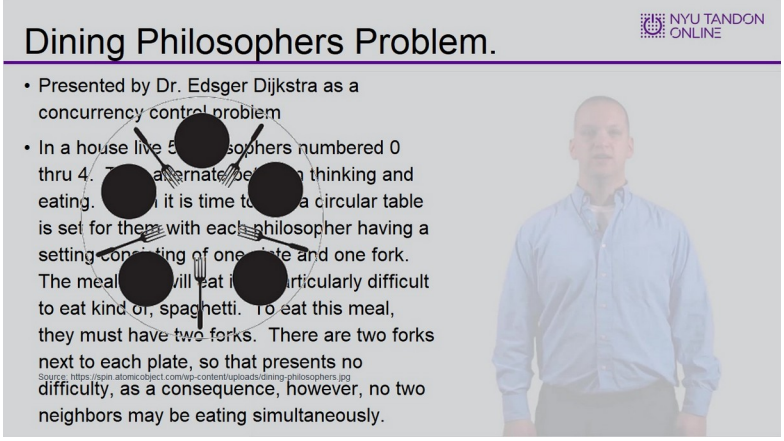
1.27 Deadlock Avoidance



1.28 Deadlock detection



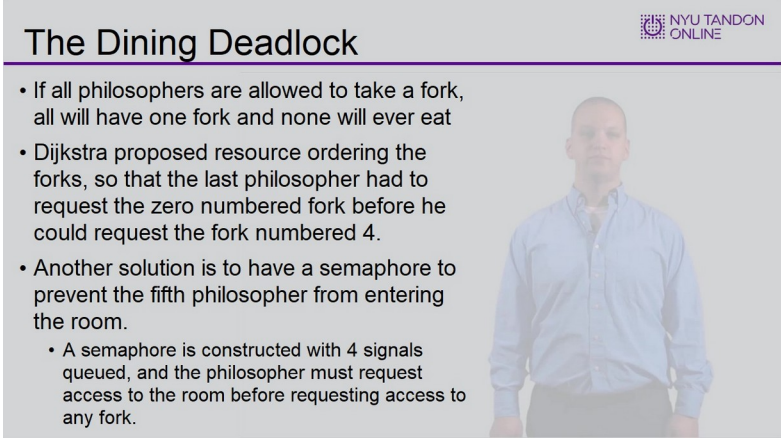
1.29 Dining Philosophers problem.



Notes:

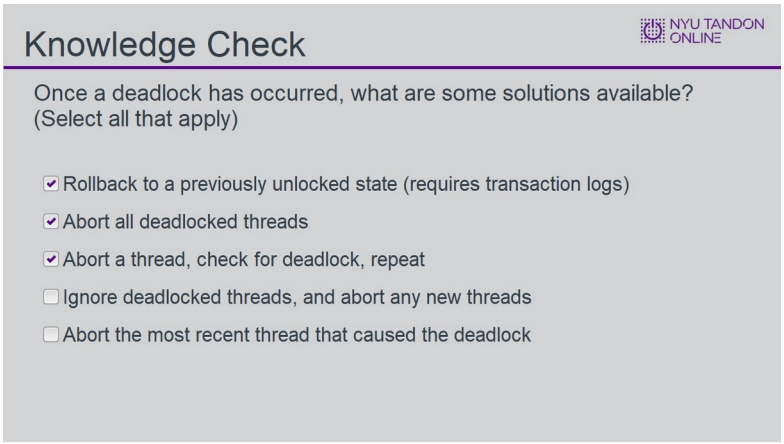
Source: https://spin.atomicobject.com/wp-content/uploads/dining-philosophers.jpg

1.30 The dining deadlock



1.31 Knowledge Check

(Multiple Response, 10 points, 4 attempts permitted)



Correct	Choice
X	Rollback to a previously unlocked state (requires transaction logs)
X	Abort all deadlocked threads
X	Abort a thread, check for deadlock, repeat
	Ignore deadlocked threads, and abort any new threads
	Abort the most recent thread that caused the deadlock

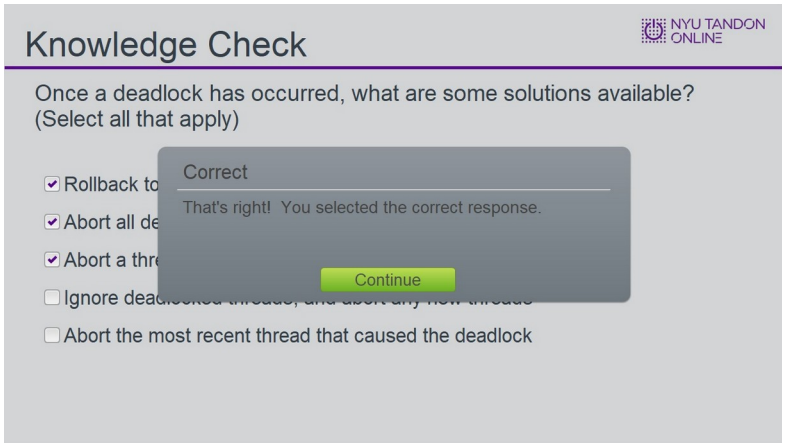
Feedback when correct:

That's right! You selected the correct response.

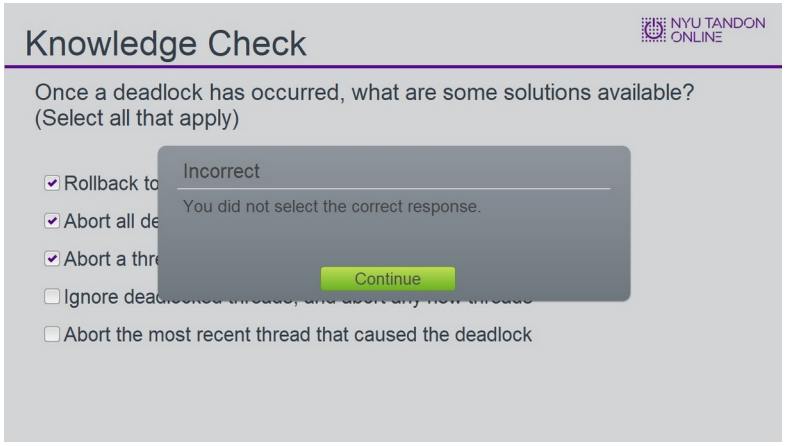
Feedback when incorrect:

You did not select the correct response.

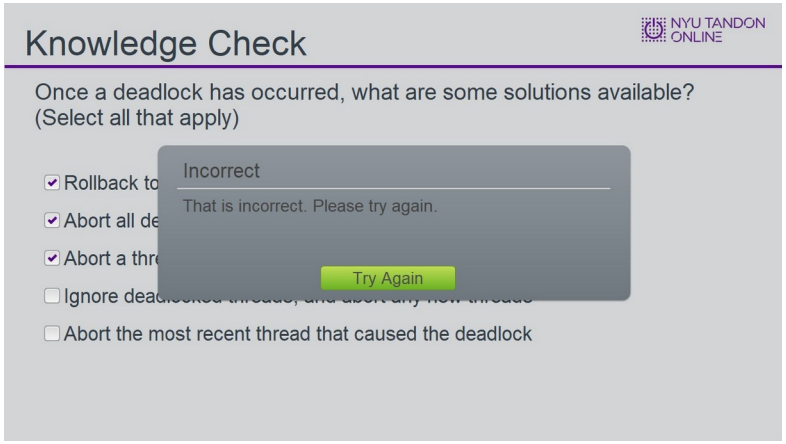
Correct (Slide Layer)



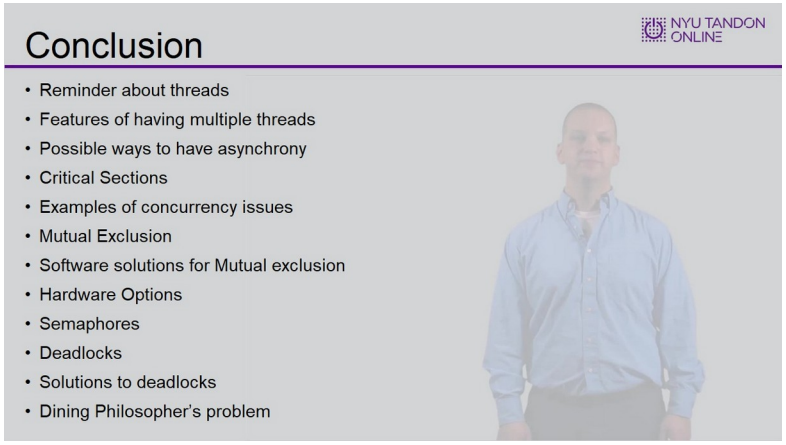
Incorrect (Slide Layer)



Try Again (Slide Layer)

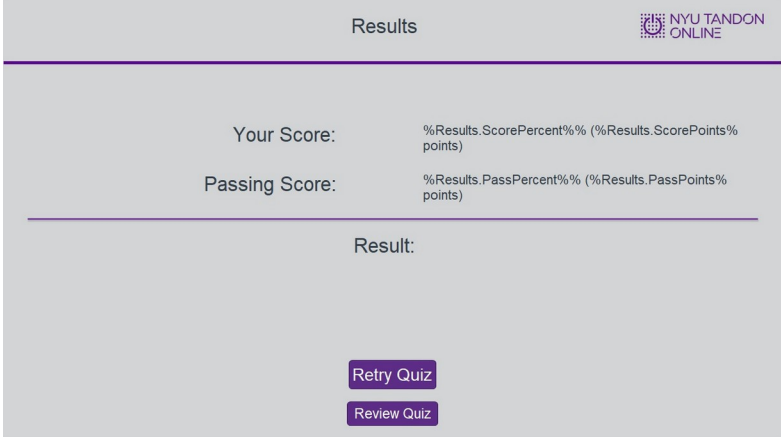


1.32 Concurrency and Deadlocks - Conclusion



1.33 Results Slide

(Results Slide, 0 points, 1 attempt permitted)

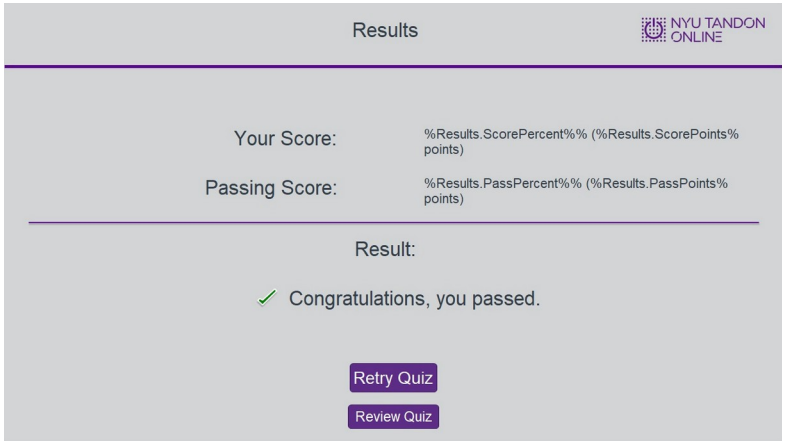


Results for
1.13 Knowledge Check
1.24 Knowledge Check
1.31 Knowledge Check

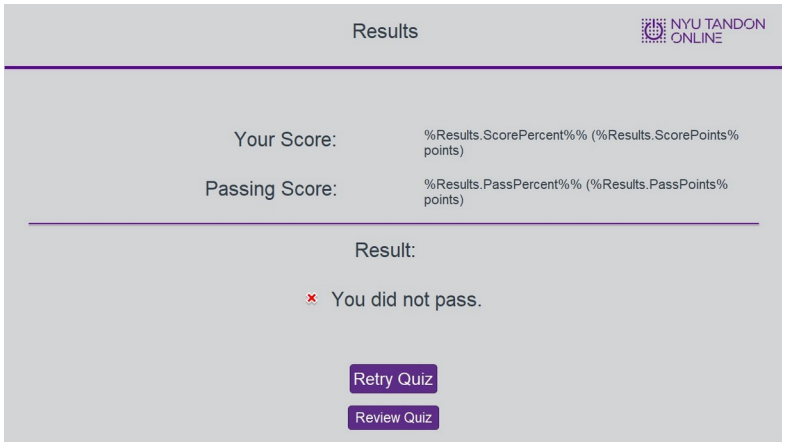
Result slide properties

Passing Score	80%

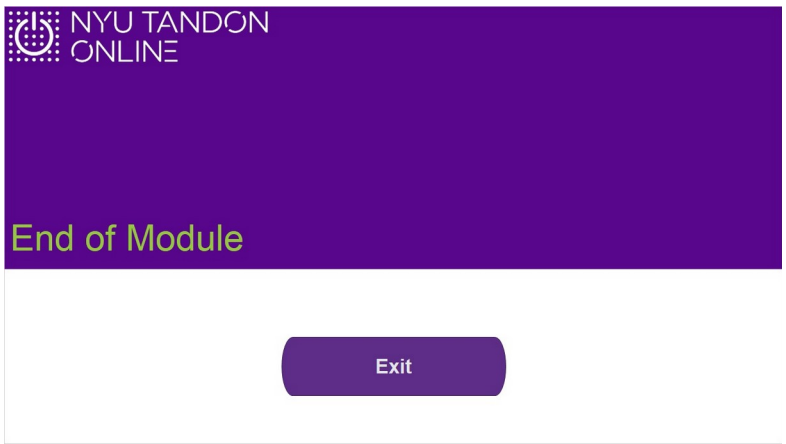
Success (Slide Layer)



Failure (Slide Layer)



1.34 End of Module



Published by Articulate® Storyline www.articulate.com