Quiz - Concurrency Bugs

Total points 44/44

Take the quiz solo, but feel free to consult a partner student, the book, the videos or other resources if needed. Re-take quiz if your score is less than 80% or if you just want some more practice.

The respondent's email (**boyapati@pdx.edu**) was recorded on submission of this form.

✓ Concurrency bugs are the most common type of bugs * 5/5

- ○ True
- ◉ False ✓

✓ Average fix time for concurrency bugs is significantly shorter than for other bugs *5/5

- ○ True
- ◉ False ✓

✓ Atomicity Violation bugs generally can be fixed with proper use of mutex locks. *5/5

- ◉ True ✓
- ○ False

✓ Lock-free synchronization is nice in theory but in practice it is never used. * 5/5

- ○ True
- ◉ False ✓
- ○ Other: .....

Which of the following types of bugs are types of Concurrency Bug? *

	Concurrency Bug	Other type of bug	Score	
Starvation	☑	☐	1/1	✓
Illegal Instruction	☐	☑	1/1	✓
Infinite Recursion	☐	☑	1/1	✓
Segmentation Fault	☐	☑	1/1	✓
Deadlock	☑	☐	1/1	✓
Null Pointer Dereference	☐	☑	1/1	✓
Data Race	☑	☐	1/1	✓
Order Violation	☑	☐	1/1	✓
Atomicity Violation	☑	☐	1/1	✓

✓ A program fails to make progress because a thread is holding something (a lock) while trying to acquire something else (another lock). This is an example of what type of concurrency bug? *5/5

- (●) Deadlock ✓
- ( ) Atomicity Violation
- ( ) Order Violation
- ( ) Starvation

✓ Ordering of lock acquisition will directly avoid what type of Concurrency bug? *5/5

- (●) Deadlock ✓
- ( ) Order Violation
- ( ) Data Race
- ( ) Atomicity Violation

✓ To avoid deadlocks an engineer proposed that all code in our project use pre-emptible locks that may be "broken" whenever an important thread (or the OS) needs to make a change to a shared data structure at high priority. This is probably a bad idea because... *5/5

- ○ the liklihood of bugs is very low
- ◉ pre-empting a running thread in the middle of its critical section leaves its memory state in a non-deterministic state. ✓
- ○ it makes the critical section much more difficult to code correctly.
- ○ there are other, simpler ways to avoid deadlock
- ○ the thread state is not shared with peer threads