Feedback — Week 1 Quiz

Help

You submitted this quiz on **Mon 12 Jan 2015 8:39 AM PST**. You got a score of **35.00** out of **39.00**. However, you will not get credit for it, since it was submitted past the deadline.

Question 1

Which of the following are motivations for concurrency described in these videos?

Your Answer		Score	Explanation
☑ Simplify program structure relative to event-driven programming	✔	1.00	
☑ Improve perceived responsiveness	✔	1.00	
☑ Make the program behave more deterministically with respect to runtime execution order	✖	0.00	
☐ Make the program easier to debug	✔	1.00	
☑ Enhance performance on multi-core platforms	✔	1.00	
Total		4.00 / 5.00	

Question Explanation

See the Section 1 Module 1 Part 1 video

Question 2

According to the videos, which of the following are reasons why purely event-driven software is hard to program?

Your Answer	Score	Explanation

☑ It's not portable across operating systems	✖	0.00
☑ It's hard to optimize its performance	✔	1.00
☑ It's behavior is non-deterministic on multi-core hardware	✖	0.00
☑ The structure of its control flow is obscured in both time and space	✔	1.00
Total		2.00 / 4.00

Question Explanation

See the Section 1 Module 1 Part 1 video

Question 3

Which of the following are examples of "accidental complexities" as described in the videos?

Your Answer		Score	Explanation
☐ Deadlocks resulting from "circular waiting"	✔	1.00	
☐ Ensuring that multiple concurrent threads don't simultaneously execute in critical sections of a program	✔	1.00	
☑ Limitations with debugging environments and debugging tools	✔	1.00	
☐ Ensuring that threads are given proper access to system resources	✔	1.00	
☑ Use of low-level application programming interfaces (APIs)	✔	1.00	
Total		5.00 / 5.00	

Question Explanation

See the Section 1 Module 1 Part 2 video

Question 4

Which of the following are examples of inherent complexities related to synchronization and scheduling presented in these videos?

Your Answer		Score	Explanation
☐ Using the POSIX Pthreads API (defined using the C programming language) to program concurrent applications	✔	1.00	
☐ Casting void pointers to whatever structure is used to pass data between a caller and callee in the Pthreads environment	✔	1.00	
☑ Ensuring applications running concurrently on an Android device don't corrupt raw contact entries in the SQLite Contacts database	✔	1.00	
☑ Scheduling the arrival and departure of airplanes based on limited resources, such as gates and runways	✔	1.00	
Total		4.00 / 4.00	

Question Explanation

See the Section 1 Module 1 Part 2 video

Question 5

Which of the following implementation elements are unique to each thread, according to the videos?

Your Answer		Score	Explanation
☑ A run-time stack	✔	1.00	
☐ The run-time heap	✔	1.00	
☐ Static data areas	✔	1.00	
☑ A program counter	✔	1.00	

Total	4.00 / 4.00

Question Explanation

See the Section 1 Module 2 Part 1 video

Question 6

Which of the following are ways that a program can give a Java Thread some code to run, according to the videos?

Your Answer		Score	Explanation
☑ Implement the Runnable interface, override its run() hook method, pass the Runnable object to the constructor of a new Thread object, and call start() on the Thread object	✔	1.00	
☑ Extend the Thread class, override its run() hook method, and call start() on an instance of the extended Thread class	✔	1.00	
☐ Extend the Thread class, override its run() hook method, and explicitly call run() from application code to start the Thread without having to call its start() method explicitly	✔	1.00	
Total		3.00 / 3.00	

Question Explanation

See the Section 1 Module 2 Part 1 video

Question 7

Which of the following statements are true according to the videos?

Your Answer		Score	Explanation
☐ The use of a volatile boolean "stop" flag automatically wakeups blocking wait(), join(), and sleep() calls	✔	1.00	

☐ The only reliable and portable way to terminate a Java Thread is to call its stop() method	✔	1.00
☐ Java the Thread interrupt() method behaves like traditional hardware & operating system interrupts, i.e., it automatically terminates a Thread regardless of what it is doing	✔	1.00
☑ If user code in a Java Thread calls wait(), join(), or sleep() these methods check if they've been interrupted and throw the InterruptedException	✔	1.00
Total		4.00 / 4.00

Question Explanation

See the Section 1 Module 2 Part 2 video

Question 8

Which of the following statements about a Java Thread's lifecycle are correct, according to the videos?

Your Answer		Score	Explanation
☐ When a Java program creates a Thread object it's initially in the Runnable state	✔	1.00	
☐ When a Java program calls sleep() the Thread transitions to the Blocked state	✔	1.00	
☑ When the Android Linux scheduler selects a Thread to execute it transitions to the Running state	✔	1.00	
☐ When a Thread's run() hook method returns the Thread transitions to the Runnable state	✔	1.00	
Total		4.00 / 4.00	

Question Explanation

See the Section 1 Module 2 Part 2 video

Question 9

Which of the following are the consequences of the Java ArrayList class implementation not being synchronized in the BuggyQueue example, according to the videos?

Your Answer		Score	Explanation
☑ If multiple threads access an ArrayList instance concurrently then it may be corrupted due to race conditions	✔	1.00	
☐ The ArrayList class should not be used in concurrent Java programs under any circumstances	✔	1.00	
☐ The ArrayList class should only be used in concurrent Java programs running on a single-core computer	✔	1.00	
Total		3.00 / 3.00	

Question Explanation

See the Section 1 Module 2 Part 3 video

Question 10

Which of the following statements about the PingPongWrong program are correct, according to the material presented in the videos?

Your Answer		Score	Explanation
☐ Although this program doesn't work properly in the Java console environment, it will work correctly on Android due to Android's multi-threaded design restrictions	✔	1.00	
☑ After the run() methods of both PlayPingPongThread objects return, the calls to their join() methods in the main Thread will also return	✔	1.00	

☑ Using Java Semaphores and CountDownLatches will make the program alternate printing "ping" and "pong" correctly, but will make the performance unacceptably slow	✖	0.00
Total		2.00 / 3.00

Question Explanation

See the Section 1 Module 2 Part 3 video