Feedback — Week 1 Quiz

Help Center

You submitted this quiz on **Thu 26 Mar 2015 12:03 PM PDT**. You got a score of **41.00** out of **41.00**.

Question 1

Which of the following are motivations for concurrency described in the videos from week 0?

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

Question Explanation

See the Section 0 Part 3 video

Question 2

According to the videos from week 0, 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	✔	1.00	
☑ The structure of its control flow is obscured in both time and space	✔	1.00	
☐ It's behavior is non-deterministic on multi-core hardware	✔	1.00	
☑ It's hard to optimize its performance	✔	1.00	
Total		4.00 / 4.00	

Question Explanation

See the Section 0 Part 4 video

Question 3

Which of the following are examples of "accidental complexities" as described in the videos from week 0?

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

Question Explanation

See the Section 0 Part 4 video

Question 4

Which of the following are examples of inherent complexities related to synchronization presented in the videos from week 0?

Your Answer		Score	Explanation
☐ Using the POSIX Pthreads API (defined using the C programming language) to program concurrent applications	✔	1.00	
☑ Scheduling the arrival and departure of airplanes based on limited resources, such as gates and runways	✔	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	
Total		4.00 / 4.00	

Question Explanation

See the Section 0 Part 4 video

Question 5

Which of the following statements about the motivation for a user-level Hardware Abstraction Layer (HAL) on Android are correct, according to the material presented in the videos in Section 1?

Your Answer		Score	Explanation
☐ The performance of running the drivers in user space is much higher than the performance of running the drivers in kernel space	✔	1.00	
☐ User space implementations of HAL capabilities have fewer restrictions in terms of available processor instructions than	✔	1.00	

kernel space implementations		
☑ Kernel device drivers are GPL'd, which requires disclosure of source code that are at odds with Android vendor desires to protect their intellectual property	✔	1.00
☑ The radio interface isn't supported in a standard way by Linux device drivers	✔	1.00
Total		4.00 / 4.00

Question Explanation

See the Section 1 Part 1 videos

Question 6

Which of the following are examples of Android concurrency frameworks, according to the videos in Section 1?

Your Answer		Score	Explanation
☐ The Android services framework, which allows computations and communication to run in the background	✔	1.00	
☐ The Java Thread class, which provides a unit of computation that runs in the context of a process.	✔	1.00	
☑ The Android "Handler Messages and Runnables" (HaMeR) framework, which allows operations to run in one or more background threads that publish their results to the UI thread	✔	1.00	
☑ The Android AsyncTask framework, which allows operations to run in one or more background threads and publish results to the UI thread without manipulating threads or handlers	✔	1.00	
Total		4.00 / 4.00	

Question Explanation

See the Section 1 Part 1 videos

Question 7

Which of the following implementation elements are unique to each thread, according to the videos from week 1?

Your Answer		Score	Explanation
☑ A program counter	✔	1.00	
☐ The run-time heap	✔	1.00	
☑ A run-time stack	✔	1.00	
☐ Static data areas	✔	1.00	
Total		4.00 / 4.00	

Question Explanation

See the Section 1 Part 2 videos

Question 8

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
☑ Extend the Thread class, override its run() hook method, and call start() on an instance of the extended Thread class	✔	1.00	
☑ 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 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 Part 4 videos

Question 9

Which of the following statements are true according to the videos in Section 1?

Your Answer		Score	Explanation
☐ The only reliable and portable way to terminate a Java Thread is to call its stop() method	✔	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	
☐ The use of a volatile boolean "stop" flag automatically wakeups blocking wait(), join(), and sleep() calls	✔	1.00	
☐ The Java Thread interrupt() method behaves like traditional hardware and operating system interrupts, i.e., it automatically terminates a Thread regardless of what it is doing	✔	1.00	
Total		4.00 / 4.00	

Question Explanation

See the Section 1 Part 3 videos

Question 10

Which of the following statements about a Java Thread's lifecycle are correct, according to the videos in Section 1?

Your Answer	Score	Explanation

☑ When the Android Linux scheduler selects a Thread to execute it transitions to the Running state	✔	1.00
☐ When a Java program calls sleep() the Thread transitions to the Blocked state	✔	1.00
☐ When a Java program creates a Thread object it's initially in the Runnable 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 Part 3 videos