Feedback — Optional Week 10 Quiz (Part 2)

Help Center

You submitted this quiz on **Mon 9 Mar 2015 3:48 PM PDT**. You got a score of **7.50** out of **10.00**. However, you will not get credit for it, since it was submitted past the deadline.

Question 1

Which of the following characterizes the key difference between the Reactor and Proactor patterns?

Your Answer		Score	Explanation
○ The Reactor pattern can be implemented in multiple programming languages, whereas the Proactor is limited to only C++ and Java			
● The Reactor handles initiative events, whereas the Proactor handles completion events	✓	1.00	
○ The Reactor pattern supports asynchronous operations, whereas the Proactor pattern supports synchronous operations			
○ The Proactor pattern uses callbacks, whereas the Reactor pattern doesn't			
Total		1.00 / 1.00	

Question 2

Which of the following are benefits of the Proactor pattern? (Check all that apply)

Your Answer		Score	Explanation
☑ Simplification of app synchronization	✓	0.25	

☑ Ease of programming, debugging, and testing	✗	0.00
☑ Efficient portability across most operating systems	✗	0.00
□ Decoupling threading from concurrency	✗	0.00
Total		0.25 / 1.00

Question 3

Which of the following reasons explain why the ACE Proactor framework has more classes than the ACE Reactor framework?

Your Answer		Score	Explanation
○ The ACE Proactor framework runs efficiently on a smaller number of operating systems platforms			
○ The ACE Proactor framework uses C++ traits more extensively			
● The use of asynchronous I/O separates operation invocation from operation completion in time and space	✓	1.00	
○ The ACE Proactor framework provides a larger set of features than the ACE Reactor framework			
Total		1.00 / 1.00	

Question 4

A proactive I/O model is harder to program than reactive and synchronous I/o models for which of the following reasons? (Check all that apply)

Your Answer		Score	Explanation
☑ There are no higher-level frameworks available to simplify the design and implementation of proactive I/O-based	✗	0.00	

applications and services			
□ It requires the use of programming languages that do not support common object-oriented constructs	✓	0.25	
☑ There's a time/space separation between asynchronous invocations and completion handling that requires tricky state management	✓	0.25	
☑ There are significant accidental complexities associated with the quality of asynchronous I/O interfaces and implementations on many operating systems	✓	0.25	
Total		0.75 / 1.00	

Question 5

Which of the following are benefits of the ACE Proactor framework? (Check all that apply)

Your Answer		Score	Explanation
☑ Provides a portable framework that runs efficiently on all operating system platforms	✗	0.00	
☑ Simplifies programming by tightly coupling I/O operations in time and space	✗	0.00	
☑ Alleviates reactive I/O bottlenecks	✓	0.25	
☑ Processes requests concurrently without the overhead of synchronous I/O and multi-threading	✓	0.25	
Total		0.50 / 1.00	

Question 6

True or false: To save space, an asynchronous completion token is not passed along with the request or response itself.

Your Answer		Score	Explanation
● False	✓	1.00	
○ True			
Total		1.00 / 1.00	

Question 7

Which of the following are NOT benefits of the Proactor path through the pattern language used to implement the JAWS web server? (Check all that apply)

Your Answer		Score	Explanation
□ Scalable concurrent processing with a minimal number of threads	✓	0.25	
□ Portability to a wide range of operating systems	✗	0.00	
□ Support for asynchronous I/O	✓	0.25	
□ Easy to understand and debug	✗	0.00	
Total		0.50 / 1.00	

Question 8

Which of the following are NOT benefits of frameworks? (check all that apply)

Your Answer		Score	Explanation
□ Reuse of implementation	✓	0.25	
☑ Steep learning curve	✓	0.25	

□ Reuse of design	✓	0.25	
□ Language-independent	✗	0.00	
Total		0.75 / 1.00	

Question 9

Which of the following are limitations of using patterns? (Check all that apply)

Your Answer		Score	Explanation
□ Subtle implementation and optimization details may be overlooked	✗	0.00	
☑ Language specific features can complicate the pattern implementation	✓	0.25	
☑ Using patterns when they are not needed incurring significantly more code	✓	0.25	
□ Shared vocabulary that enhances communication	✓	0.25	
Total		0.75 / 1.00	

Question 10

What is a design pattern?

Your Answer		Score	Explanation
○ A repetitive design present in some programming languages			
● A reusable solution to a common problem in a given context	✓	1.00	
○ A recurring programming language element			

Total	1.00 / 1.00