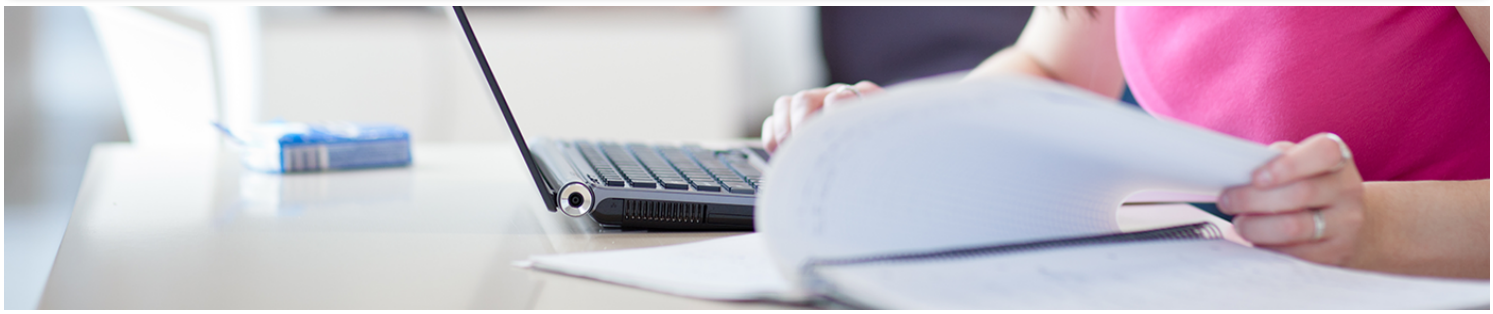
Due Feb 18 at 9:51pm **Points** 25 **Questions** 20
Available Feb 18 at 9:05pm - Feb 18 at 10:05pm about 1 hour **Time Limit** 45 Minutes

Instructions

Quiz 1



Home

Modules

Resources

Answer all questions.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	32 minutes	25 out of 25

ⓘ Correct answers will be available Feb 18 at 10:05pm - Feb 18 at 11:30pm.

Score for this quiz: **25** out of 25
Submitted Feb 18 at 9:38pm
This attempt took 32 minutes.

Question 1 **1 / 1 pts**

Concurrency is needed in many systems for the following reasons:

- Concurrency reduces the load on the CPU and memory
- Increasing application performance and exploiting multicore processors
- Concurrency makes it easier to write programs
- Concurrent systems are easier to test

Correct!

Question 2

1 / 1 pts

Which of the below statements is false? Choose One.

- Concurrency is fundamental to a distributed system
- Distributed systems can handle multiple simultaneous requests
- Events can happen on different nodes at the same time in a distributed system
- The order of events is guaranteed in a distributed system

Question 3

1 / 1 pts

Which of the following statements is true? Choose One.

- (selected) Multicore systems are essential for distributed systems
- It is not possible to scale distributed systems
- All distributed systems are synchronous
- A Client browser is essential for a distributed system

Question 4

1 / 1 pts

Which of the following statements is true?

- It is not possible for threads to share the same address space
- Different processes can share the same address space
- Both threads and processes can share the same address space
- (selected) Different threads can share the same address space

Question 5

1 / 1 pts

Replicating data makes a system more resilient and available. What major issues must a system with replicated data handle in order to keep the replicas consistent?

- Concurrent updates only.
- Concurrent updates, network partitions, and sharks biting through cables.
- (selected) Concurrent updates and network partitions.

Network partitions only.

Correct!

Question 6

1 / 1 pts

Making a scalable system highly available requires:

- Scaling out the system's components and handling inevitable failures.
- Using commercial cloud-based services which can make sure a system is always available at no cost.
- Scaling up the system.
- Scaling out the system and making sure the client application code is error free and does not crash.

Correct - Scaling out and handling failures are key to high availability.

Question 7

1 / 1 pts

If a program has a race condition, which of the following are true?

- Running the program with identical inputs can produce different results.
- All of the above
- The program is hard to debug.
- The program sometimes produces the correct results.

Correct!

Question 8

1 / 1 pts

In the following, class NamingThread simply prints out the string passed to its constructor and exits. Given the code below, what will the program output be?

```
public class ThreadStartOrderExample {

    public static void main(String arg[]) {

        Thread th1 = new Thread (new NamingThread("Pep the Great"))
;

        Thread th2 = new Thread (new NamingThread("Mourinho the
tool"));

        Thread th3 = new Thread (new NamingThread("grrr")) ;

        System.out.println ("Ready to roll");

        th1.start();

        th2.start();

        th3.start();

        System.out.println ("main thread exiting " +
Thread.currentThread());

    }
```

```
}
```

Ready to roll

grrr

Mourinho the tool

Pep the Great

main thread exiting thread0

Ready to roll

Pep the Great

Mourinho the tool

grrr

main thread exiting thread0

It is impossible to predict.

Ready to roll

Mourinho the tool

grrr

Pep the Great

main thread exiting thread0

Correct!

Question 9

1 / 1 pts

Is this class thread safe?

```
public class Factorizer extends GenericServlet implements Servlet {
    public void service(ServletRequest req, ServletResponse resp) {
```

```
    BigInteger i = extractFromRequest(req);
    BigInteger[] factors = factor(i);
    encodeIntoResponse(resp, factors);
  }
```

- It is only threadsafe if it is called by one client at a time.
- It sure is! Perfectly threadsafe.
- It is only threadsafe if run on a different server than the calling classes.
- No - multiple threads calling this class will cause a race condition.

Correct. This class is stateless and hence threadsafe.

Question 10

2 / 2 pts

Look at the code below. Which server threading model does this code implement?

```
public class ThreadedServer {
   public static void main(String[] args) throws IOException {
      ServerSocket socket = new ServerSocket(80);
      while (true) {
         final Socket connection = socket.accept();
         Runnable task = new Runnable() {
            public void run() {
               handleRequest(connection);
```

```
            }
        };
        new Thread(task).start();
    }
  }
  private static void handleRequest(Socket connection) {
    // request-handling logic here
  }
}
```

- (●) thread per client request
- ( ) Varying size thread pool
- ( ) Fixed size thread pool
- ( ) thread per database connection

Correct!

Question 11

2 / 2 pts

Which of the following are disadvantages of the thread per request model? Choose all that apply.

- [ ] It is not able to effectively exploit multicore CPUs due to thread affinity.
- [ ] It results in considerably more network traffic.

- [x] Each thread uses additional memory, which can lead to OutofMemory exceptions under high loads.

- [x] It incurs overheads on every request for thread creation and destruction.

Correct!

Question 12

1 / 1 pts

What are the challenges that come with concurrency? Choose all that apply.

- [ ] Threads
- [x] Race conditions
- [x] Deadlocks
- [ ] Processes

Question 13

1 / 1 pts

Which of the following statements is **NOT** true about locks in concurrent systems?

- Placing a lock around a resource serializes access to that resource.

All threads must first acquire a lock and then release the lock when they have completed their operations.

Locks must always be acquired in a deterministic order.

A lock can only be held by one thread.

Correct!

Question 14

2 / 2 pts

[Select] are more lightweight than [Select] because they share the same address space and code and data segments

Answer 1:

threads

Answer 2:

processes

Correct!

Question 15

2 / 2 pts

deadlock occurs when threads acquire locks in an order such that none of the threads can make further progress.

Answer 1:

deadlock

Answer 2:

none

Correct!

Question 16

1 / 1 pts

Race conditions are caused by _____ behavior.

non-deterministic

Question 17

1 / 1 pts

Load balancers will always distribute requests randomly to a group of replicated servers

- True
- False

Question 18

1 / 1 pts

The only effective way to grow a database is to scale up the system to a highly reliable, multi-CPU and disk server

- ○ True
- ◉ False

Question 19

1 / 1 pts

How is it possible to break the deadlock in the dining philosophers problem is?

- ◉ Impose total ordering on the acquisition of forks
- ○ Never put down the fork a philosopher has picked up
- ○ Each philosopher can pick up fork on their right first
- ○ All of the above

Question 20

2 / 2 pts

When the wait() method of a Thread is invoked, what does the Thread then do?

- ○ Waits for a notification (notify()) of a free resource
- ○ waits to acquire a lock on an object

waits for completion of another thread

All of the above

Quiz Score: **25** out of 25