



Review Test Submission: Quiz 10 - Concurrency

User	Kaidi He
Course	201503 Advanced Software Paradigms_CSCI_6221_11
Test	Quiz 10 - Concurrency
Started	11/11/15 6:11 PM
Submitted	11/11/15 6:47 PM
Due Date	11/11/15 6:50 PM
Status	Completed
Attempt Score	65 out of 100 points
Time Elapsed	36 minutes out of 39 minutes
Results Displayed	All Answers

Question 1

0 out of 5 points

Assume the following method is properly synchronized and called from a thread A on an object B:

```
wait(2000);
```

After calling this method, when will the thread A become a candidate to get another turn at the CPU?

Answers: After thread A is notified, or after two seconds.

Two seconds after thread A is notified.

After the lock on B is released, or after two seconds.

Two seconds after lock B is released.

Question 2

5 out of 5 points

Given the following Java code, what is the output if you create an object as `new A().run("3")` ?

```
class A extends Thread
{
    public A()
    {
        System.out.print("1");
    }
    public void run()
    {
        System.out.print("2");
    }
    public void run(String s)
    {
        System.out.print(s);
    }
}
```

Answers: 12

13

12

123

Question 3

10 out of 10 points

Regarding Java, which of the following about sleep() and yield() are true?

Answers: **All of the above**

sleep() and yield() are static methods in Thread class

sleep() forces the current thread to go for sleep for specified period

yield() voluntarily relinquishes CPU to other threads

Question 4

0 out of 10 points

Given the following Java class, which of the following line of code is suitable to start a thread ?

```
class X implements Runnable {    public static void main(String args[])    {        /* Missing code */    }    public void run() {} }
```

Answers: Thread t = new Thread(X); t.start();

X run = new X(); Thread t = new Thread(run); t.start();

Thread t = new Thread(); x.run();

Thread t = new Thread(X);

Question 5

5 out of 5 points

For the following Java class, choose the correct thread class:

```
public class Test12
{
    public static void main(String[] args)
    {
        Thread1 t = new Thread1();
        t.start();
    }
}
```

Answers: **class Thread1 extends Thread {**
public void run() {} }

class Thread1 implements Thread {
public void run() {} }

class Thread1 implements Runnable {
public void run() {} }

class Thread1 extends
Runnable {
public static void run() {} }

Question 6

5 out of 5 points

When deadlock occurs, assuming that only two program units are causing the deadlock, one of the involved program units should be gracefully terminated, thereby allowed the other to continue.

Answers: **True**
False

Question 7

5 out of 5 points

Java constructors can be specified to be synchronized.

Answers: True

False

Question 8

10 out of 10 points

Given the following Java code, which of the statement(s) are true considering class Account?

```
class Account
{
    private int balance = 10000;
    public void debit(int amt)
    {
        if (amt < 0)
            return;
        synchronized(this)
        {
            balance -= amt;
        }
    }
}
```

Answers: Two concurrent threads cannot access *amt*

all answer are wrong

Two concurrent threads cannot access either *balance* or *amt*

Two concurrent threads cannot access *balance*

Question 9

5 out of 5 points

```
public class MyRunnable implements Runnable {    public void run()    {        // some code here    } }
```

which of these will create and start this thread?

Answers: new MyRunnable().start();

new Thread(new MyRunnable()).start();

new Runnable(MyRunnable).start();

new Thread(MyRunnable).run();

Question 10

0 out of 10 points

Which of the following will directly stop the execution of a Thread?

Answers: notify()

exits synchronized code

notifyall()

wait()

Question 11

5 out of 5 points

Consider the following Java class

```

class Even {
    private int n = 0;
    public int next(){
        // POST-CONDITION: next is always even
        ++n;
        Thread.sleep(500);
        ++n;
        return n;
    }
}

```

If multiple threads access Even, post-conditions may not be kept

Answers: **True**

False

Question 12

0 out of 10 points

Assuming your programming in Java, how would you prevent two threads accessing a shared resource?

Answers: all answers are correct

Use volatile variable

Use synchronized block

Use synchronized method

Question 13

5 out of 5 points

Assume the following Java class

```

class Cell {
    private long value;
    synchronized long getValue() { return value; }
    synchronized void setValue(long v) {value = v; }
    synchronized void swapValue(Cell other)
    {
        long t = getValue();
        long v = other.getValue();
        setValue(v);
        other.setValue(t);
    }
}

```

If two generic instances of Cell, say a and b, are concurrently invoking swapValue, the program may block indefinitely because a needs b's lock and vice versa.

Answers: **True**

False

Question 14

10 out of 10 points

Which of the following is true about sleep() and wait()?

Answers: **wait() releases the lock while the thread is waiting**

wait() does not release the lock while the thread is waiting

sleep() releases the lock while the thread is sleeping

sleep() does not release the lock while the thread is sleeping

Wednesday, November 11, 2015 6:50:42 PM EST

← OK