

[Try Another Quiz](#)**Question: 1** Which two are valid constructors for Thread?

1. Thread(Runnable r, String name)
2. Thread()
3. Thread(int priority)
4. Thread(Runnable r, ThreadGroup g)
5. Thread(Runnable r, int priority)

**Your Answer:** 2 and 5 ❌**Correct Answer:** 1 and 2**Description:** (1) and (2) are both valid constructors for Thread.**Question: 2** Which method registers a thread in a thread scheduler?**Your Answer:** start(); ✅**Correct Answer:** start();**Description:** The start() method causes this thread to begin execution; the Java Virtual Machine calls the run method of this thread.**Question: 3** class X implements Runnable

```
{
public static void main(String args[])
{
/* Missing code? */
}
public void run() {}
}
```

Which of the following line of code is suitable to start a thread ?


**Your Answer:** Thread t = new Thread(X); t.start(); ❌**Correct Answer:** X run = new X(); Thread t = new Thread(run); t.start();**Description:** None**Question: 4** Which three guarantee that a thread will leave the running state?

1. yield()
2. wait()
3. notify()
4. notifyAll()
5. sleep(1000)
6. aLiveThread.join()
7. Thread.killThread()

**Your Answer:** 2, 5 and 6 ✅**Correct Answer:** 2, 5 and 6**Description:** None**Question: 5** Which will contain the body of the thread?**Your Answer:** run(); ✅**Correct Answer:** run();**Description:** The run() method to a thread is like the main() method to an application. Starting the thread causes the object' run method to be called in that separately executing thread.

**Question: 6** Which two of the following methods are defined in class Thread?


1. start()
2. wait()
3. notify()
4. run()
5. terminate()

**Your Answer:** 1 and 4 

**Correct Answer:** 1 and 4

**Description:** 1 and 4 Only start() and run() are defined by the Thread class.


**Question: 7** Which method must be defined by a class implementing the java.lang.Runnableinterface?

**Your Answer:** public void run() 

**Correct Answer:** public void run()

**Description:** because in an interface all methods are abstract by default therefore they must be overridden by the implementing class. The Runnableinterface only contains 1 method, the void run() method therefore it must be implemented.


**Question: 8** What is the name of the method used to start a thread execution?

**Your Answer:** start(); 

**Correct Answer:** start();

**Description:** The start() method causes this thread to begin execution; the Java Virtual Machine calls the run method of this thread.

**Question: 9** Which of the following will not directly cause a thread to stop?

**Your Answer:** sleep() 


**Correct Answer:** notify()

**Description:** notify() - wakes up a single thread that is waiting on this object's monitor.

**Question: 10** 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?

**Your Answer:** Two seconds after thread A is notified. 

**Correct Answer:** After thread A is notified, or after two seconds.

**Description:** Either of the two events (notification or wait time expiration) will make the thread become a candidate for running again.

**Question: 11** Which of the following will directly stop the execution of a Thread?

**Your Answer:** exits synchronized code 

**Correct Answer:** wait()


**Description:** wait() causes the current thread to wait until another thread invokes the notify() method or the notifyAll() method for this object.

**Question: 12** public class MyRunnable implements Runnable

```
{  
    public void run()  
}
```

```
// some code here  
}  
}  
}
```


which of these will create and start this thread?

**Your Answer:** new Thread(new MyRunnable()).start(); 

**Correct Answer:** new Thread(new MyRunnable()).start();

**Description:** Because the class implements Runnable, an instance of it has to be passed to the Thread constructor, and then the instance of the Thread has to be started.

**Question: 13** Which class or interface defines the wait(), notify(),and notifyAll() methods?


**Your Answer:** Thread 

**Correct Answer:** Object

**Description:** The Object class defines these thread-specific methods.

**Question: 14** Which three are methods of the Object class?

1. notify();
2. notifyAll();
3. isInterrupted();
4. synchronized();
5. interrupt();
6. wait(long msec);
7. sleep(long msec);
8. yield();

**Your Answer:** 1, 2, 4 


**Correct Answer:** 1, 2, 6

**Description:** (1), (2), and (6) are correct. They are all related to the list of threads waiting on the specified object. (3), (5), (7), and (8) are incorrect answers. The methods isInterrupted() and interrupt() are instance methods of Thread.

The methods sleep() and yield() are static methods of Thread.

D is incorrect because synchronized is a keyword and the synchronized()construct is part of the Java language.

**Question: 15** Which cannot directly cause a thread to stop executing?

**Your Answer:** Calling read() method on an InputStream object. 

**Correct Answer:** Calling notify() method on an object.

**Description:** notify() wakes up a single thread that is waiting on this objects monitor.

Finish

Tweet

Like 126K

G+