

## Java – Learn From Home

### Quiz – Chapter 5

1. What is the name of the method used to start a thread execution?

- A. `init();`
- B. `start();`
- C. `run();`
- D. `resume();`

Answer: Option B

Explanation:

Option B is Correct. The `start ()` method causes this thread to begin execution; the Java Virtual Machine calls the `run` method of this thread.

Option A is wrong. There is no `init()` method in the `Thread` class.

Option C is wrong. The `run ()` method of a thread is like the `main ()` method to an application. Starting the thread causes the object's `run` method to be called in that separately executing thread.

Option D is wrong. The `resume ()` method is deprecated. It resumes a suspended thread

2. Which two are valid constructors for `Thread`?

`Thread(Runnable r, String name)`

`Thread()`

`Thread(int priority)`

`Thread(Runnable r, ThreadGroup g)`

`Thread(Runnable r, int priority)`

- A. 1 and 3
- B. 2 and 4
- C. 1 and 2

D. 2 and 5

Answer: Option C

Explanation:

(1) And (2) are both valid constructors for Thread.

(3), (4), and (5) are not legal Thread constructors, although (4) is close. If you reverse the arguments in (4), you'd have a valid constructor

3. Which three are methods of the Object class?

notify();

notifyAll();

isInterrupted();

synchronized();

interrupt();

wait(long msecs);

sleep(long msecs);

yield();

A. 1, 2, 4

B. 2, 4, 5

C. 1, 2, 6

D. 2, 3, 4

Answer: Option C

Explanation:

(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.

4. 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?

- A. Thread t = new Thread(X);
- B. Thread t = new Thread(X); t.start();
- C. X run = new X(); Thread t = new Thread(run); t.start();
- D. Thread t = new Thread(); x.run();

Answer: Option C

Explanation:

Option C is suitable to start a thread

5. Which two of the following methods are defined in class Thread?  
start()  
wait()  
notify()  
run()  
terminate()

- A. 1 and 4
- B. 2 and 3
- C. 3 and 4
- D. 2 and 4

Answer: Option A

Explanation:

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

(2) And (3) are incorrect because they are methods of the Object class. (5) Is incorrect because there's no such method in any thread-related class.

6. Which three guarantee that a thread will leave the running state?

yield()  
wait()  
notify()  
notifyAll()  
sleep(1000)  
aLiveThread.join()  
Thread.killThread()

- A. 1, 2 and 4
- B. 2, 5 and 6
- C. 3, 4 and 7
- D. 4, 5 and 7

Answer: Option B

Explanation:

(2) Is correct because wait() always causes the current thread to go into the object's wait pool.

(5) Is correct because sleep() will always pause the currently running thread for at least the duration specified in the sleep argument (unless an interrupted exception is thrown).

(6) Is correct because, assuming that the thread you're calling join() on is alive, the thread calling join() will immediately block until the thread you're calling join() on is no longer alive.

(1) Is wrong, but tempting. The yield() method is not guaranteed to cause a thread to leave the running state, although if there are runnable threads of the same priority as the currently running thread, then the current thread will probably leave the running state.

(3) And (4) are incorrect because they don't cause the thread invoking them to leave the running state.

(7) is wrong because there's no such method

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

- A. `wait()`
- B. `notify()`
- C. `notifyall()`
- D. exits synchronized code

Answer: Option A

Explanation:

Option A is correct. `wait()` causes the current thread to wait until another thread invokes the `notify()` method or the `notifyAll()` method for this object.

Option B is wrong. `notify()` - wakes up a single thread that is waiting on this object's monitor.

Option C is wrong. `notifyAll()` - wakes up all threads that are waiting on this object's monitor.

Option D is wrong. Typically, releasing a lock means the thread holding the lock (in other words, the thread currently in the synchronized method) exits the synchronized method. At that point, the lock is free until some other thread enters a synchronized method on that object. Does entering/exiting synchronized code mean that the thread execution stops? Not necessarily because the thread can still run code that is not synchronized. I think the word directly in the question gives us a clue. Exiting synchronized code does not directly stop the execution of a thread

8. Which method must be defined by a class implementing the `java.lang.Runnable` interface?

- A. `void run()`
- B. `public void run()`
- C. `public void start()`
- D. `void run(int priority)`

Answer: Option B

Explanation:

Option B is correct because in an interface all methods are abstract by default therefore they must be overridden by the implementing class. The Runnable interface only contains 1 method, the void run() method therefore it must be implemented.

Option A and D are incorrect because they are narrowing the access privileges i.e. package(default) access is narrower than public access.

Option C is not method in the Runnable interface therefore it is incorrect.

9. Which will contain the body of the thread?

- A. run();
- B. start();
- C. stop();
- D. main();

Answer: Option A

Explanation:

Option A is Correct. The run() method to a thread is like the main() method to an application. Starting the thread causes the object's run method to be called in that separately executing thread.

Option B is wrong. The start() method causes this thread to begin execution; the Java Virtual Machine calls the run method of this thread.

Option C is wrong. The stop() method is deprecated. It forces the thread to stop executing.

Option D is wrong. Is the main entry point for an application

10. Which method registers a thread in a thread scheduler?

- A. run();
- B. construct();
- C. start();
- D. register();

Answer: Option C

Explanation:

Option C is correct. The start () method causes this thread to begin execution; the Java Virtual Machine calls the run method of this thread.

Option A is wrong. The run () method of a thread is like the main () method to an application. Starting the thread causes the object's run method to be called in that separately executing thread.

Option B is wrong. There is no construct () method in the Thread class.

Option D is wrong. There is no register() method in the Thread class