**QUIZIZZ** **Worksheets**

Name

**Session 3**

Class

Total questions: 8

Date

Worksheet time: 4mins

1.
```
public synchronized void inc(long delta) {
    this.value += delta;
}
```

Which object is synchronized?

a)  We can't tell from just this

b)  Nothing

c)  this (the object the method belongs to)

2.  When a thread leaves a synchronized block, other threads that want to enter it automatically try to do so.

a)  True

b)  False

3.  In which state is a thread after we call .start()?

a)  NOT RUNNABLE

b)  RUNNABLE

c)  NEW

d)  TERMINATED

4.  When does a thread leave the RUNNABLE state without entering the terminated state?

a)  After creating it, before we call .start()

b)  After .run() finished executing

c)  When we call .notRunnable()

d)  When it becomes blocked or waiting

5.  If a thread enters a NOT RUNNABLE state because it .waits() on an object, how does it go back to the RUNNABLE state?

a)  If .notify() or .notifyAll() is called

b)  After a certain amount of time it automatically happens

c)  It keeps checking if the object is no longer locked until it succeeds

d)  If we call .start() on it

6.

```
public class Foo {
    public void synchronized f() { … }
    public void synchronized g() { … f(); … }
}
```

Can a thread call synchronized() on the same object multiple times?

a) Yes

b) No, this will make the program run forever

c) No, this causes an exception

7. How can a thread become the owner of an object's monitor (lock)?

a) By executing a method/block synchronized on the object

b) Threads can't own monitors

c) By calling .wait() on the object

8. To call .wait() or .notify() on an object, a thread has to own the monitor of that object.

a) False

b) True

**Answer Keys**

1. c) this (the object the method belongs to)

2. a) True

3. b) RUNNABLE

4. d) When it becomes blocked or waiting

5. a) If .notify() or .notifyAll() is called

6. a) Yes

7. a) By executing a method/block synchronized on the object

8. b) True