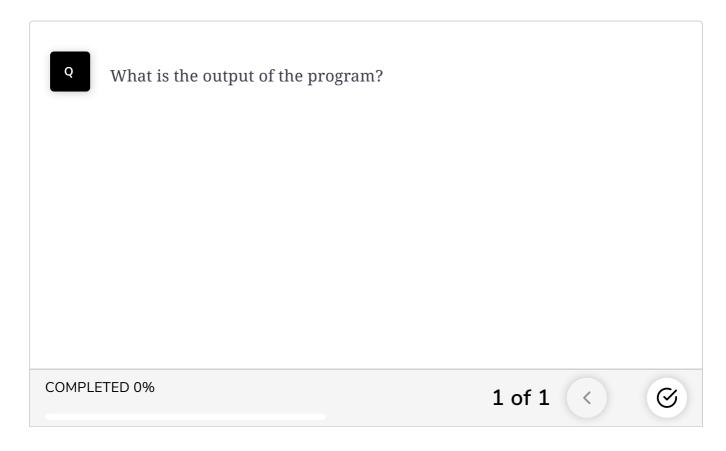
Quiz 2

Questions relating to the Threading API are covered in this lesson.

Question # 1 Consider the code snippet below: Thread.new do Thread.stop() end "Main thread exiting"











[]

Question # 2

Consider the snippet below:

```
mutex = Mutex.new
Thread.new do
 mutex.lock()
 raise "Ka Boom"
# wait for child thread to exit
sleep(2)
puts("Is mutex locked #{mutex.locked?}")
mutex.lock()
```

What is the outcome of running the above snippet?

COMPLETED 0%

1 of 1 <



```
mutex = Mutex.new

Thread.new do
    mutex.lock()
    raise "Ka Boom"
end

# wait for child thread to exit
sleep(2)
puts("Is mutex locked #{mutex.locked?}")
mutex.lock()
```







[]

Question # 3

Consider the snippet below:

```
mutex = Mutex.new

# lock once
mutex.lock()

# lock twice
mutex.lock()

puts "Main thread exiting"
```

Q

What is the outcome of running the above snippet?





```
mutex = Mutex.new
                                                                                      G
# lock once
mutex.lock()
# lock twice
mutex.lock()
puts "Main thread exiting"
```

Question # 4

Consider the snippet below:

```
mutex = Mutex.new()
def doSomethingImportant(mtx)
  begin
    mtx.lock()
    raise "Ka Boom"
    mtx.unlock()
  rescue RuntimeError => e
    # swallow the exception
Thread.new do
  # invoke do something important
 doSomethingImportant(mutex)
  # simulate other work the thread undertakes
 sleep(1000)
end
```

```
# wait for child to start execution
sleep(3)

# attempt to acquire mutex
puts "Main thread attempting to acquire lock"
mutex.lock()

puts "Main thread exiting"
```

What is the outcome of running the program?

COMPLETED 0%

1 of 1 〈 ⓒ

```
mutex = Mutex.new()

def doSomethingImportant(mtx)

begin
    mtx.lock()
    raise "Ka Boom"
    mtx.unlock()

rescue RuntimeError => e
    # swallow the exception
    end
end

Thread.new do
    # invoke do something important
```

```
# simulate other work the thread undertakes

sleep(1000)
end

# wait for child to start execution
sleep(3)

# attempt to acquire mutex
puts "Main thread attempting to acquire lock"
mutex.lock()

puts "Main thread exiting"
```







[]

Question # 5

In the previous snippet, we change the

```
def doSomethingImportant(mtx)

begin
  mtx.synchronize {
    raise "Ka Boom"
  }

rescue RuntimeError => e
  # swallow the exception
  end
end
```

Q

What is the outcome of running the program?

COMPLETED 0%

1 of 1 <





```
mutex = Mutex.new()
                                                                                        C)
def doSomethingImportant(mtx)
 begin
   mtx.synchronize {
     raise "Ka Boom"
  rescue RuntimeError => e
   # swallow the exception
 end
end
Thread.new do
 # invoke do something important
 doSomethingImportant(mutex)
 # simulate other work the thread undertakes
 sleep()
end
# wait for child to start execution
sleep(3)
# attempt to acquire mutex
puts "Main thread attempting to acquire lock"
mutex.lock()
puts "Main thread exiting"
```







