



Quiz 2

Test what you have learnt so far.

Question # 1

Consider the snippet below:

```
from multiprocessing import Value, Lock

lock = Lock()
pi = Value('d', 3.1415, lock=lock)

lock.acquire()
print(pi.value)
lock.release()
```

Q What will be the output of running the above snippet?



A) Deadlock



B) 3.1415 is printed on the console



C) Error is raised



Submit Answer

Reset Quiz ↺

```
from multiprocessing import Value, Lock
```



```
lock = Lock()  
pi = Value('d', 3.1415, lock=lock)
```

```
lock.acquire()  
print(pi.value)  
lock.release()
```



Question # 2

Consider the below snippet:



```
from multiprocessing import Process, Value, Semaphore
import multiprocessing

def child_process(item1, item2, sem):
    sem.acquire()
    print("{0} {1}".format(str(item1.value), str(item2)), flush=True)

if __name__ == '__main__':
    multiprocessing.set_start_method('fork')

    # initialize both variables to 1
    var1 = Value('I', 1)
    var2 = 1

    sem = Semaphore(0)

    process = Process(target=child_process, args=(var1, var2, sem))
    process.start()

    var1.value = 7
    var2 = 7
    sem.release()

    process.join()
```

Q What is the output of the above program?

**A) 1 and 1****B) 1 and 7****C) 7 and 1****D) 7 and 7****Submit Answer****Reset Quiz**

```
from multiprocessing import Process, Value, Semaphore
import multiprocessing
```



```
def child_process(item1, item2, sem):
    sem.acquire()
    print("{0} {1}".format(str(item1.value), str(item2)), flush=True)
```

```
if __name__ == '__main__':
    multiprocessing.set_start_method('fork')
```

```
    # initialize both variables to 1
    var1 = Value('I', 1)
    var2 = 1
```

```
    sem = Semaphore(0)
```

```
    process = Process(target=child_process, args=(var1, var2, sem))
    process.start()
```

```
    var1.value = 7
    var2 = 7
    sem.release()
```

```
    process.join()
```





Question # 3

Consider the snippet below, where the developer is trying to use the Semaphore class from the threading module.

```
from multiprocessing import Process
from threading import Semaphore
import multiprocessing

def child_process():
    sem.acquire()
    print("Hi, I am the child")

if __name__ == '__main__':
    multiprocessing.set_start_method('fork')

    sem = Semaphore(0)

    process = Process(target=child_process, )
    process.start()

    sem.release()
    process.join()
```

Q What would be the output of running the above program?



A) Runs to completion successfully



B) Program hangs



C) An error is raised

Submit Answer

Reset Quiz ↺

```
from multiprocessing import Process
from threading import Semaphore
import multiprocessing

def child_process():
    sem.acquire()
    print("Hi, I am the child")

if __name__ == '__main__':
    multiprocessing.set_start_method('fork')

    sem = Semaphore(0)

    process = Process(target=child_process, )
    process.start()

    sem.release()
    process.join()
```



Question # 4

The start method in the previous question is changed to "spawn"



Q What would be the output of the program with start method changed to spawn?

☐ A) Runs to completion successfully

☐ B) Program hangs

☐ C) An error is raised

Submit Answer

Reset Quiz ↺

```
from multiprocessing import Process
from threading import Semaphore
import multiprocessing
```

```
def child_process():
    sem.acquire()
    print("Hi, I am the child")
```

```
if __name__ == '__main__':
    multiprocessing.set_start_method('spawn')

    sem = Semaphore(0)

    process = Process(target=child_process)
    process.start()

    sem.release()
    process.join()
```





Question # 5

In the above program what fixes should the developer apply to make the program work?

Q

- ☐ A) set start method to forkserver and use multiprocessing.thread
- ☐ B) set start method to spawn and use multiprocessing.thread
- ☐ C) set start method to fork and use multiprocessing.thread

Submit Answer

Reset Quiz ↺





```
from multiprocessing import Process, Semaphore
import multiprocessing

def child_process():
    sem.acquire()
    print("Hi, I am the child")

if __name__ == '__main__':
    multiprocessing.set_start_method('fork')

    sem = Semaphore(0)

    process = Process(target=child_process)
    process.start()

    sem.release()
    process.join()
```

[← Back](#)[Next →](#)[Quiz 1](#)[Quiz 3](#)[Mark as Completed](#)[Report an Issue](#)[Ask a Question](#)

(https://discuss.educative.io/tag/quiz-2__multiprocessing__python-concurrency-for-senior-engineering-interviews)