

Self-Diagnostic Quiz 11

Please comply with the [NUS Plagiarism Policy](https://myportal.nus.edu.sg/studentportal/student-discipline/all/docs/NUS-Plagiarism-Policy.pdf) (<https://myportal.nus.edu.sg/studentportal/student-discipline/all/docs/NUS-Plagiarism-Policy.pdf>) and [NUS Code of Student Conduct](https://studentconduct.nus.edu.sg/wp-content/uploads/NUS-Code-of-Student-Conduct.pdf) (<https://studentconduct.nus.edu.sg/wp-content/uploads/NUS-Code-of-Student-Conduct.pdf>) (section A on academic integrity). [NUS Libraries' Academic Integrity Essentials](https://libguides.nus.edu.sg/new2nus/acadintegrity) (<https://libguides.nus.edu.sg/new2nus/acadintegrity>) summarises these in an accessible manner.

- Due 9 Apr at 23:59
- Points 15
- Questions 15
- Available 7 Apr at 14:00 - 9 Apr at 23:59
- Time limit None
- Allowed attempts Unlimited

Take the quiz again

Attempt history

	Attempt	Time	Score
LATEST	<u>Attempt 1</u>	43 minutes	8.32 out of 15

⚠ Correct answers will be available 10 Apr at 0:01 - 4 May at 0:00.

Score for this attempt: 8.32 out of 15

Submitted 7 Apr at 23:27

This attempt took 43 minutes.



PartialQuestion 1

0.5 / 1 pts

Which of the following code snippet ensures that "world" is always printed after "hello"?

(select all correct answers)



```
System.out.println("hello");  
System.out.println("world");
```



```
System.out.println("hello");  
new Thread(() -> { System.out.println("world"); }).start();
```



```
new Thread(() -> { System.out.println("hello"); }).start();  
System.out.println("world");
```



```
new Thread(() -> { System.out.println("hello"); }).start();  
new Thread(() -> { System.out.println("world"); }).start();
```



```
new Thread(() -> { System.out.println("hello"); });  
new Thread(() -> { System.out.println("world"); });
```



Question 2

1 / 1 pts

Consider the following snippet:

```
new Thread(() -> {  
    for (int i = 0; i < 3; i += 1) {  
        System.out.print("***");  
    }  
}).start();  
  
new Thread(() -> {  
    for (int i = 0; i < 3; i += 1) {  
        System.out.print("@");  
    }  
}).start();
```

Which of the following might be printed? (select all correct answers)

- ☒ ***@@@
- ☒ *@*@@
- ☐ ***@@*
- ☐ *@@*@@
- ☒ **@@*@@

⋮

IncorrectQuestion 3

0 / 1 pts

Consider the following snippet:

```
Supplier<Integer> s = () -> fib(100);  
CompletableFuture cf = CompletableFuture.supplyAsync(s);
```

When is the CompletableFuture `cf` considered as complete?

- ☐ After the method call `CompletableFuture::supplyAsync` returns.
- ☐ After the method call `fib(100)` returns.
- ☐ After the method call `s.get()` returns.
- ☐ After the reference of a new CompletableFuture is assigned to `cf`.
- ☒ After the method call `cf.join()` returns.

⋮

Question 4

1 / 1 pts

Consider the following snippet:

```
CompletableFuture<Integer> cf = CompletableFuture.completedFuture(fib(100));
```

When is the CompletableFuture `cf` considered as complete?

- ☒ After the method call `CompletableFuture::completedFuture` returns.
- ☐ After the method call `fib(100)` returns.
- ☐ After the reference of a new CompletableFuture is assigned to `cf`.

☐ After the method call `cf.join()` returns.



IncorrectQuestion 5

0 / 1 pts

Consider the function below:

```
CompletableFuture<Integer> createAsync() {  
    Supplier s1 = () -> fib(100);  
    return CompletableFuture.supplyAsync(s1); // Line A  
}
```

and the following invocation:

```
Supplier<CompletableFuture<Integer>> s2 = () -> createAsync();  
CompletableFuture<CompletableFuture<Integer>> cf = CompletableFuture.supplyAsync(s2); // Line B
```

When is the CompletableFuture cf considered as complete?

☐ After the method call `CompletableFuture::supplyAsync` at Line A returns.

☒ After the method call `CompletableFuture::supplyAsync` at Line B returns.

☐ After the method call `fib(100)` returns.

☐ After the method call `s1.get()` returns.

☐ After the method call `s2.get()` returns.

☐ After both method calls `s1.get()` and `s2.get()` returns.

☐ After the method call `cf.join()` returns.



PartialQuestion 6

0.75 / 1 pts

Which of the following code snippet ensures that "world" is always printed after "hello"?

(select all correct options)



```
CompletableFuture  
    .runAsync(() -> System.out.println("hello"))  
    .thenRun(() -> System.out.println("world")).join();
```



```
CompletableFuture  
    .runAsync(() -> System.out.println("hello"))  
    .thenRunAsync(() -> System.out.println("world")).join();
```



```
CompletableFuture  
    .runAsync(() -> {  
        System.out.println("hello");  
        System.out.println("world");  
    }).join();
```



```
CompletableFuture  
    .runAsync(() -> {})  
    .thenRunAsync(() -> System.out.println("hello"))  
    .thenRunAsync(() -> System.out.println("world")).join();
```



```
CompletableFuture<Void> cf = CompletableFuture.runAsync(() -> {});  
cf.thenRunAsync(() -> System.out.println("hello"));  
cf.thenRunAsync(() -> System.out.println("world")).join();
```

Question 7

1 / 1 pts

Consider the following snippet:

```
CompletableFuture.runAsync(() -> {  
    for (int i = 0; i < 3; i += 1) {  
        System.out.print("*");  
    }  
}).thenRunAsync(() -> {  
    for (int i = 0; i < 3; i += 1) {  
        System.out.print("@");  
    }  
}).join();
```

Which of the following might be printed?

(select all correct answers)

- ☒ ***@@@
- ☐ **@@*@
- ☐ *@*@@
- ☐ ***@@*
- ☐ *@*@@@



PartialQuestion 8

0.67 / 1 pts

Consider the following program:

```
class Main {  
    public static void main(String[] args) {  
        CompletableFuture.runAsync(() -> {  
            for (int i = 0; i < 3; i += 1) {  
                System.out.print("*");  
            }  
        }).thenRunAsync(() -> {  
            for (int i = 0; i < 3; i += 1) {  
                System.out.print("@");  
            }  
        }); // no join  
    }  
}
```

Which of the following might be printed? (select all correct answers)

- ☒ *
- ☒ **@
- ☒ ***@@
- ☒ ***@@@
- ☐ *@**@@



Question 9

1 / 1 pts

Consider the following program:

```
class Main {  
    public static void do(int i) {  
        // doSomething  
        System.out.print(i);  
    }  
  
    public static void main(String[] args) {  
        CompletableFuture.allOf(  
            CompletableFuture.runAsync(() -> do(1)),  
            CompletableFuture.runAsync(() -> do(2)))  
            .thenRun(() -> do(3))  
            .join();  
    }  
}
```

Which of the following might be printed?

(select all correct answers)

- ☒ 123
- ☒ 213
- ☐ 132
- ☐ 231
- ☐ 12
- ☐ 13
- ☐ 312
- ☐ 321
- ☐ 3

⋮

PartialQuestion 10

0.4 / 1 pts

Consider the following program:

```
class Main {  
    public static void do(int i) {  
        // doSomething  
        System.out.print(i);  
    }  
  
    public static void main(String[] args) {  
        CompletableFuture.anyOf(  
            CompletableFuture.runAsync(() -> do(1)),  
            CompletableFuture.runAsync(() -> do(2)))  
            .thenRun(() -> do(3))  
            .join();  
    }  
} // <- changed
```

Which of the following might be printed?

(select all correct answers)

- ☒ 123

- ☒ 213
- ☒ 132
- ☒ 231
- ☐ 12
- ☒ 13
- ☒ 312
- ☒ 321
- ☒ 3
- ☐ ⋮

The next 4 questions concern the RecursiveTask below:

```
class Task extends RecursiveTask<Integer> {
    private int x;

    Task(int x) {
        this.x = x;
    }

    @Override
    protected Integer compute() {
        if (x >= 4) {
            return x;
        }
        Task t1 = new Task(2 * x);
        Task t2 = new Task(2 * x + 1);
        t1.fork();
        t2.fork();
        return t2.join() + t1.join();
    }
}
```

When we call `new Task(1).compute()`, it creates two subtasks, denoted as Task(2) and Task(3). Task(2) creates two more subtasks Task(4) and Task(5); while Task(3) creates two more subtasks Task(6) and Task(7).

⋮

IncorrectQuestion 11

0 / 1 pts

Suppose there is only one worker thread and it runs

```
new Task(1).compute()
```

After Task(3) is forked, what is the content of the deque for this worker thread? (The content is listed from head to tail, with the head first and the tail last).

- ☐ 1 2 3
- ☐ 3 2
- ☒ 2 3
- ☐ 2
- ☐ 3
- ☐ ⋮

IncorrectQuestion 12

0 / 1 pts

After the `join` for Task(3) is invoked, what is the content of the deque for this worker thread? (The content is listed from head to tail, with the head first and the tail last).

- ☐ 2 4 5 6 7
- ☐ 2 7 6
- ☐ 7 6 2
- ☒ 2 6 7
- ☐ 6 7
- ☐ 7 6



IncorrectQuestion 13

0 / 1 pts

After the `join` for Task(2) is invoked, what is the content of the deque for this worker thread? (The content is listed from head to tail, with the head first and the tail last).

- ☐ 7 6 5 4
- ☐ 4 5 6 7
- ☐ 5 4
- ☐ 4 5
- ☐ 3 4 5
- ☐ 3 5 4
- ☒ it is empty



Question 14

1 / 1 pts

Suppose that there is a second worker thread who tries to steal from the first worker thread, while it is running Task(3). Which task would it steal?

- ☒ Task(2)
- ☐ Task(4)
- ☐ Task(5)
- ☐ Task(6)
- ☐ Task(7)



Question 15

1 / 1 pts

Suppose the task above is the only task stole by the second worker thread from the first worker thread. The first worker thread completes all of its tasks in its deque and wishes to steal some work from the second worker thread. Which of the following task is likely to be stolen by the first worker thread?

- ☒ Task(4)
- ☐ Task(5)
- ☐ Task(6)
- ☐ Task(7)

Quiz score: 8.32 out of 15