Course
CS-E4110
Course materials
Your points
MyCourses
Zulip Chat

**This course has already ended.**
**The latest instance of the course can be found at: Concurrent Programming: 2023**

« 3 Semaphore Blocking Queue -- Concurrent data structures based on lo... Course materials 5 Feedback »

CS-E4110 / Round 3 - Synchronization primitives / 4 Scala (JVM) Memory model and Data Races. / Submission 5

Assignment description | My submissions (5/5)

# Scala (JVM) Memory model and Data Races.

**1. *Which are among the reasons a thread might not immediately see results of an operation in another thread?*** 1 / 1

- ☑ **A Compiler may reorder instructions otherwise than stated in the original source code.**
- ☐ Out of thin air execution in JVM.
- ☑ **Processors may process instructions in parallel or out of order.**
- ☑ **Caches may vary the order in which writes happen to main memory.**
- ☑ **Threads may choose to store results of computation in their local cache before committing them later.**

✓ Correct!

**2. *The JMM defines a partial ordering called happens-before on all actions within a program. Choose the ones that belong to this happens-before rule?*** 1 / 1

- ☑ **An unlock on a monitor lock happens-before every subsequent lock on that same monitor lock.**
- ☐ A write to a volatile field happens-before every subsequent read of any field.
- ☑ **A call to Thread.start on a thread happens-before every action in the started thread.**
- ☑ **Any action in a thread happens-before any other thread detects the thread has terminated.**
- ☑ **If a happens-before b, and b happens-before c, then a happens-before c.**

✓ Correct!

**3. *Which are possible results of executing the following program under JVM??*** 0 / 1

```
//shared variables
int x = 0, y = 0;
int a = 0, b = 0;
```

```
// thread 1 (t1)
a = 1;
x = b;
```

```
// thread 2 (t2)
b = 1;
y = a;
```

```
// results
t1.start(); t2.start();
t1.join(); t2.join();
println("( "+ x + "," + y + ")");
```

- ☐ (1,0).
- ☑ **(0,1).**
- ☐ (1,1).
- ☑ **(0,0).**

Imagine all possible reorderings and interleavings.

✗ Incorrect

**4. *Which ones are true regarding the following program?*** 2 / 2

```
//shared variables
int x = 0, y = 0;
```

```
// thread 1 (t1)
r1 = x;
if (r1 != 0) y = 1;
```

```
// thread 2 (t2)
r2 = y;
if (r2 != 0) x = 1;
```

- ☐ JMM allows the line `r1 = x;` to see the write of `x = 1`.
- ☑ **The line `r1 = x;` being able to see the write of `x = 1` is happens-before consistent according to JMM.**
- ☑ **The program is correctly synchronized according to JMM.**
- ☐ The line `r2 = y;` being able to see the write of `y = 1` is sequentially consistent according to JMM.

✓ Correct!

Submit

Earned points
8 / 10

## Exercise info

**Assignment category**
Multiple choice questionnaires

**Your submissions**
5 / 5

**Deadline**
Monday, 22 November 2021, 14:00

**Late submission deadline**
Monday, 29 November 2021, 14:00 (-30%)

**Total number of submitters**
43

## Submission info

**Submitted on**
Sunday, 14 November 2021, 16:21:20

**Status**
Ready

**Grade**
8 / 10

**Submitters**
Binh Nguyen

« 3 Semaphore Blocking Queue -- Concurrent data structures based on lo... Course materials 5 Feedback »