

This homework is individual. Please read Clean Code Chapter 13 first. Checkout the **concurrency-start** branch from gitlab. Total points is 14.

1. (2 points) Producer-consumer model. If the queue is full, who will be waiting, the producer or the consumer?

The producer

2. (2 points) Reader-writer model. Explain what is *starvation*.

Starvation is when the readers are taking in more information than the writers are processing, which means not all the information gets processed.

3. (3 points) Race Condition. Show how you use **synchronized** in **void run()** to prevent the race condition. You MUST use an object as a lock instead of using a **synchronized** method.

```
public void run() {
    for (int i = 0; i < 500; i++) {
        synchronized (cnt) {
            cnt.update();
        }
    }
}
```

4. (3 points) Dining philosopher. Show how you use **synchronized** in **void run()** to prevent the dead lock. You should use an object as a lock instead of using a **synchronized** method.

```
public void run() {
    synchronized (Philosopher.class) {
        think();
        pickLeft();
        pickRight();
        eat();
        downRight();
        downLeft();
    }
}
```

5. (4 points) Adam and Betty share a bank account. The pseudo code (looking suspiciously like Java though) of the bank system is listed below:

