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.

<u>Starvation is when the readers are taking in more information than the writers are processing.</u>
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();
        }
    }
}</pre>
```

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:

```
class Bank {
   int balance;
   //...
   void withDraw(int amount) {
    int oldBalance = balance;
   int newBalance = oldBalance - amount;
   balance = newBalance;
}
```

Assume each statement is *atomic* but withDraw is not. The starting balance of the account is \$100. If both Adam and Betty withdraw \$80, the balance can be \$20 at the end. Demonstrate how this happens in the following table. Write down the statements being executed at a given "round". Only one statement can be executed in each "round". Write down the value of balance in the first column before executing the statement in the current row.

Balance	Adam	Betty
\$100	set oldBalance ← balance	
\$100	set newBalance ← oldBalance - \$80	set oldBalance ← balance
\$100	set balance ← newBalance	set newBalance ← oldBalance - \$80
\$20	balance ← \$20	set balance ← newBalance
\$20		balance ← \$20

One more table if the first one is not enough

Balance	Adam	Betty