**Question1:**

**Process:**

an executing program

**Thread:**

Individual and separate unit of execution and it’s part of a process

**Examples**:

web services, video game, FTP download, common browsers, multi-thread operation file

**Advantages:**

Easier to program (1 thread per task); can provide better performance (thread only runes when needed; no polling to decide what to do); multiple threads can share resources; utilize multiple processors if available.

**Task:**

Tasks are structured on threads, which means that tasks are ultimately thrown to threads for execution; Tasks and threads are not in a one-to-one relationship.

**Yield:**

current thread gives up processor so another of equal priority can run (if none of equal priority, it runs again)

**States of Java Threads with Wait/Notify:**

new (just created but not started), runnable (created, started, and able to run), blocked (created and started but unable to run because it is waiting for some event to occur), dead (thread has finished or been stopped)

Wait: to suspend a thread

Notify: to wake a thread up

**Question3:**

r,q,t

**Question7:**

First, create an Input object ip and the parameter index and input were initialized. Then, the lock object is created. Two threads are created afterwards which are t1 and t2 contains a MyThread object with two parameters which are ip and lock and a string “Thread1” and “Thread2”. When the object was created, the parameters ip and lock in the object was initialized to the same as the ip and lock in Caller class. Then, when the threads are started and joined, the run method in the MyThread class is called. Parameter index is initialized to int -1. Then the while loop is executed. Index of the ip object is 0, so input[0] is printed at first. However, synchronized makes the threads work at different time, which makes the number that was printed from 1 to 15 occurs only once for each. For the reason that the threads are working randomly, the output may be thread 1 and thread 2 intermixed but the number under each thread should increase monotonously.

**Question9:**

Hello main

Hello Thread-0

Done!