

Metropolitan Stat University, Saint Paul, Minnesota  
ICS 462 Operating Systems  
Assignment 6 Part 2

Goal: To learn to program using concurrent data structures and compare disk scheduling algorithms

Problem: In this part of the assignment, you will create three Java classes.

1. **Producer.** This class either extends **Thread** or implements **Runnable**. The constructor receives the reference to a **Requests** object passed in as parameter and stores it in a field of this class. In the **run()** method, the class generates 25 cylinder requests and puts each one of them into the **Requests** object. Between each successive pair of requests, the thread sleeps for a random period of time up to an integer value stored in the final variable **DELAY\_BETWEEN\_REQUESTS** defined in the **Driver** class (see below). (Use the random number generator in Java.) The thread should display the cylinder values it generated.
2. **Consumer.** This class either extends **Thread** or implements **Runnable**. The constructor receives the reference to a **Requests** object passed in as parameter and stores it in a field of this class. In the **run()** method, the class calls the **get()** method of the same **Requests** object that the producer uses, to get all the 25 cylinder requests, possibly through multiple calls. If there are no requests, the consumer waits (pass the value **true**). Make sure that the thread sleeps between successive requests and that this sleep time is the integer value stored in **SLEEP\_TIME**, defined in the **Driver** class (see below). The consumer should print the returned value from the **get()** method calls.
3. **Driver.** Creates a **Requests** object and starts up a single **Producer** and a single **Consumer** thread to work on the **Requests** object.

You need to submit the implementation (the source code for the above three classes and **Requests**) by 11:59 PM on July 16. I will accept late submissions until 11:59 PM on July 17. There will be a 10% penalty for late submissions. I will post my implementation on July 18.