Project 3—Producer-Consumer Problem (in Java)
(**Due date: 11/13/2008/Thursday**)

Your name:	Date:

In Chapter 6, we have presented a semaphore-based solution (with part of the pseudo code) to the Producer-Consumer problem using a bounded buffer. In the class on 10/24/2008/Friday we also introduced several Java code samples for dealing this problem, by using Java's Thread synchronization features (the sample code is available from the course website). Based on these examples, in this project, we will enhance functionality of the synchronization illustrated in them. The requirements are specified in the following:

- The buffer is a **Circular/Bounded Buffer** with 5 elements (integers);
- The Producer can produce only ***one item*** (integer) instead of 10 in the example. In the Test/Driver's main(), you need to create 15 threads of Producer. The control of sleep() time may be put either in Producer or in main();
- The Consumer can take only ***one item*** (integer) instead of 10 in the example. In the Test/Driver's main(), you need to create 15 threads of Consumer. The control of sleep() time may be put either in Producer or in main().
- You **need to use Executors, ExecutorService*, Lock, Condition,*** and ***ReentrantLock*** for the synchronization instead of using Synchonized method/statements.
- For displaying processing and results, you need follow the examples illustrated in the sample code: each time, when a thread is setting/getting (trying to set or get) an item to/from the buffer, the states of the buffer need to be displayed: Read/Write indexes, Full/Empty, Waiting/Reading/Writing, What (the integer number) is being read/written…(GUI for displaying is NOT required).

After you have tested your project code, modify/save your code (*.java files **only**) in different directories for submission according to the following requirements:

1. Under subdirectory "project31src", your project code should illustrate:
    a. First creating 15 Consumer threads without any sleep time;
    b. Then creating 15 Producer threads without any sleep time;
    c. The results under this condition.
2. Under subdirectory "project32src", your project code should illustrate:
    a. First creating 15 Consumer threads with a random sleep time (1-5 seconds) for each thread;
    b. Then creating 15 Producer threads with a random sleep time (1-5 seconds) for each thread;
    c. The results under this condition.
3. Under subdirectory "project33src", your project code should illustrate:
    a. First creating 15 Producer threads with a random sleep time (1-5 seconds) for each thread;
    b. Then creating 15 Consumer threads with a random sleep time (1-5 seconds) for each thread;
    c. The results under this condition.
4. Under subdirectory "project34src", your project code should illustrate:
    a. In a loop with 15 iterations, first creating one Consumer threads with a random sleep time (1-5 seconds), and then creating one Producer threads with a random sleep time (1-5 seconds);
    b. The results under this condition.

5. (**Bonus only**) Under subdirectory "project35src", your project code should illustrate:
    a. The Producer will produce **two** items (integers) each time and the Consumer will still take **only one** item each time;
    b. In a loop with 15 iterations, first creating one Consumer threads with a random sleep time (1-5 seconds), and then creating one Producer threads with a random sleep time (1-5 seconds);
    c. The results under this condition.
6. (**Bonus only**) Under subdirectory "project36src", your project code should illustrate:
    a. The Producer will produce **only one** items (integers) each time and the Consumer will still take **two** item each time;
    b. In a loop with 15 iterations, first creating one Consumer threads with a random sleep time (1-5 seconds), and then creating one Producer threads with a random sleep time (1-5 seconds);
    c. The results under this condition.

How to Grade:

- The score for the above 1, 2, 3, and 4 parts is 100 points. If you have completed one of them, you will get 50 points; if two, you get 75; if three, you get 90.
- The bonus for both 5 and 6 is 60 points; but you will get 40 for one of them.
- **In addition to above compressed source java files, a readme file (PDF or Doc or DOCX format) is required for your submission. Check the following on how to submit your project.**

==========How To Submit—Read Carefully, Pease!!==========

1. Create a directory "**project3_YourLastName**" (you must use this format for the directory name for this project; **Use Your Last Name. For example, if your** last name is Smith, you should create directory with the name of "project3_Smith"
2. Create "**project31src**" … "**project36src**" subdirectories under "project3_YourLastName" directory.
3. Under these subdirectories, you can put ONLY java files (source) files. This should be clean and comprehensive—that is, I will javac *.java and I can test your code.
4. If you have used some IDE, you can compress the package files in other subdirectories than the above six ones and tell me how to run in the **readme** file.
5. A "**readme**" file is required for the project write-up that tells how to compile in which IDE (not required if not having used any IDE but a simple command line), result screenshots (one for each), … keep this readme simple!
    a. This "readme" must reside in the "**project3_YourLastName**" dir in the format of .txt, .pdf, or .doc/docx.
6. Compress the "**project3_YourLastName**" dir and its contents into a zipped/rar-ed file with same name.
7. Submit the compressed file to the instructor by email.