Summary:
This project will continue to assist you in the use of the Java development environment. This project will introduce the concept of thread programming. You can use the NetBeans/JAVA development environment available on the Internet from Sun Microsystems and should consult a standard Java reference to assist you in the completion of this assignment. Note: that this program is command line-based and not GUI centric, so use of the Netbeans environment is not required.

Description:
You are to create a Java program called threads1.java program. The program should print your complete name, student ID, current date and time when the main program is executed, and a text string that identifies this course “INFS601-1 Spring 2006, Project Assignment 2” as a first action.

The main program will then create multiple threads and perform some actions based upon those threads. Note the program needs to properly clean-up prior to completion.

The number of threads that the program creates is based upon a random number, which is between one and a maximum number of threads, which is specified as a command line parameter. Each thread is to have a unique identifier. Each thread when created executes to perform it’s function. Each thread is to execute for a given number of iterations as specified by a command-line parameter. Each thread’s function when running is to provide appropriate information to be used by the main program so that the main program can determine the execution sequence of all threads created.

When each thread completes all of its iterations, then that thread finishes, otherwise the thread suspends execution for a random time amount of time as defined as follows:

\[
\text{thread-suspend-period} - \text{a random number between 1 and } \#\text{max-thread-suspend which is expressed in milliseconds as a command-line parameter}
\]

Use the Math.random() to generate random numbers.

NOTE: Each thread is to suspend for a different random amount of time.

Prior to starting the first thread, the main program indicates on standard output, the total number of threads that will be created, all command line parameters given with their descriptions including the thread-suspend-period used. When all threads have completed their execution iterations, the main program issues an execution trace of the threads using the thread ID to indicate the execution order of the threads with one thread ID per printed line.
The final action that the main program performs is to issue a statement on the standard output device indicating the “main program “is complete”.

Your program will be graded with the Java 5.0 JDK (JSEE/JRE). If you use any non-standard packages you do so at your own risk. In addition, you program must use standard thread methods that are available within the JAVA environment. Ensure that your program properly handles any exceptions that could be generated when your program is executing.

The command line format of the program is defined as follows:

```
java  threads1  #max-threads  #max-thread-suspend  #number-iterations
```

where the command line parameters are:
- `#max-threads` - the upper bounds on the number of threads that are created which is > 0 and ≤ 20 (for testing purposes)
- `#number-iterations` – is the number of iterations for thread execution
- `#max-thread-suspend` – is the maximum amount to suspend a thread in milliseconds

**NOTE:** all parameters represent integer numbers (no decimals).

Your program will be graded upon properly executing when the TA uses various command line parameters as test inputs.

**Programming Practices Requirements:**

Your source code is to be properly commented and follow standard programming practices. Points will be deducted at the discretion of the TA for poorly commented and written code.

JAVA code guidelines can be found here:  
[
]

How to Write Doc Comments for the Javadoc  
[
http://java.sun.com/j2se/javadoc/writingdoccomments/
]

Java Programming Style Guidelines  
[
http://geosoft.no/development/javastyle.html
]

**Submission Information:**

Submit your source code, executable code, build/compile information, screen capture or log (script) of the output from the program’s execution, readme file, and notes (if any) in a single zip file to the TA before the assigned TA due date and time. Ensure that you receive an acknowledgement from the TA that your project submission has been received on time.

At a minimum, submit your source code and a screen capture / log of the program’s execution to the instructor on the assigned due date.

15 February 2006