UEEN3433/3123 TCP/IP NETWORK APPLICATION DEVELOPMENT

PRACTICAL 1

1. Write a multithreaded program that performs the following five tasks concurrently:

   - Prints the multiples of six 200 times.
   - Prints the character ‘U’ 175 times.
   - Prints the string “Meow” 125 times.
   - Prints a random number 100 times.

   You are required to define the necessary task classes. Each of the task class must implements the `Runnable` interface.

2. Modify your program in Q1 so that your task classes extend the `Thread` class instead.

3. Modify your program in Q1 to use *thread pools*.

4. Write a multithreaded program that performs the following four tasks concurrently:

   - Appends the string “Meow” to a file.
   - Appends the string “Quack” to the same file.
   - Appends the string “Woof” to the same file.
   - Appends the string “Moo” to the same file.

   Ensure that each thread can only append to the file when the file is not in use by another thread by using *synchronized method*.

5. Modify your program in Q4 to use *synchronized statement*.

6. Modify your program in Q4 to use *explicit locks*.

7. Write a multithreaded program that performs the following two tasks concurrently.

   - Writes a string to a file.
   - Reads the content of the same file, and empty the file after reading.

   Ensure that the file can be written only when it is empty, and the file can be read only when it has content.

NOTE: For Questions 4 to 7, please refer to http://www.mkyong.com/java/how-to-write-to-file-in-java-bufferedwriter-example/ for tutorials on using the ***BufferedWriter*** class in Java for file output.