

Thread Worksheet I

1. What is the relationship that exists between a program, a process, and threads?
2. Does the Thread start method starts a thread immediately? Discuss.
3. When does a thread finish?
4. Which of the following are shared by threads in a process?
 - a. Heap
 - b. Stack
 - c. Program Counter (PC)
 - d. Files
5. Define a method that creates a two dimensional array with random values. The method takes as parameter the dimensions of the array and returns the array. Using this method, define code that computes the sum of elements in a two-dimensional array using one thread per row. The result of each thread should be left in one entry of a one-dimensional array (e.g., the first array entry has the sum computed by the first thread, the second array entry the sum computed by the second thread, etc.). Once all threads have finished, we can compute the total sum by adding the values in the one-dimensional array. Compare the efficiency of using the multi-threaded approach against one that relies only on one thread to compute the sum of all elements in the array. Try different array sizes.

For time computation use the following code fragment:

```
long startTime = System.nanoTime(); // starting the clock

// TASK YOU WANT TO TIME

long elapsed = System.nanoTime() - startTime; // stopping the clock
System.out.println("Time elapsed in nanoseconds: " + elapsed);
System.out.println("Time elapsed in seconds: " + (double) elapsed / NANoseconds_IN_A_SEC);
```

Do you see an advantage to using threads? Discuss.

6. Implement a size() method for a binary tree that relies on a thread to find the size of the left subtree and a second thread to find the size of the right subtree.