

Threads Homework Part I

Due Wednesday, April 16, 2008 at 9:00am in class

1. Why would you design a program to use multiple threads?
2. Below are two implementations of a method that waits for t milliseconds. What are the tradeoffs between these two designs?

<pre>// Code A: public void pause(int t) { long t1 = System.currentTimeMillis() + t; while (System.currentTimeMillis() < t1); }</pre>	<pre>// Code B: public void pause (int t) { try { Thread.sleep(t); } catch (InterruptedException e) {} }</pre>
--	--

3. Below are two pieces of code that are executing once on separate threads. Variable balance is an int whose initial value is 500. What are the possible final values for balance? Assume for this problem that each line of code runs uninterrupted (which is not generally true!) and that the program is executing on a single processor core.

<pre>// Thread 1: int x = balance; x = x + 100; balance = x;</pre>	<pre>// Thread 2: int y = balance; y = y - 100; balance = y;</pre>
---	---

4. Repeat problem 1 with the following code and an initial balance of 50.

<pre>// Thread 1: if (balance > 0) { int x = balance; x = x + 100; balance = x; }</pre>	<pre>// Thread 2: int y = balance; y = y - 100; balance = y; y = balance; y = y - 100; balance = y;</pre>
---	---