```java
package thread_join;
import java.util.ArrayList;

public class Thread_join {

	static ArrayList<Thread> tname = new ArrayList<Thread>();
	static boolean done = false;
	/*
	 * Constructor for Thread_join, it is responsible for firing off the
	 * threads
	 */
	Thread_join (int numT1, int numT2) {
		for (int i=0; i<numT1; i++) {
			tname.add(new Thread (new T1()));
		}
		for (int i=0; i<numT2; i++) {
			tname.add(new Thread(new T2()));
		}
		for (int i=0; i<tname.size(); i++) {
			tname.get(i).start();
		}
	}
	class T1 implements Runnable{
		public void run() {
			while (!done){
				try {
					Thread.sleep(1000);
				} catch (InterruptedException e) {
					System.out.println("T1 interrupted.");
				}
			}
			System.out.println("T1 thread exiting.");
		}
	}
	class T2 implements Runnable{
		public void run() {
			while (!done){
				try {
					Thread.sleep(1000);
				} catch (InterruptedException e) {
					System.out.println("T2 interrupted.");
				}
			}
			System.out.println("T2 thread exiting.");
		}
	}
	public static void main(String args[]) {
		/*
		 * Call the constructor to fire off the threads
		 */
		new Thread_join (2,2);
		/*
		 * Check to see if the threads are alive
		 */
		for (int i=0; i<tname.size(); i++)
			System.out.println ("Thread #" + i + " is alive: " + tname.get(i).isAlive());
		/*
		 * Sleep for 5 seconds
		 */
		try {
			Thread.sleep(5000);
		} catch (InterruptedException e) {
			System.out.println ("Main thread interrupted");
		}
		/*
		 * Set done to true and join the outstanding threads
		 */
		done = true;
		try {
			for (int i=0; i<tname.size(); i++)
				tname.get(i).join();
		} catch (InterruptedException e) {
			System.out.println("Main thread Interrupted");
		}
		/*
		 * Show that the threads are gone
		 */
		for (int i=0; i<tname.size(); i++)
			System.out.println ("Thread #" + i + " is alive: " + tname.get(i).isAlive());

		System.out.println("Main thread exiting.");
	}
}
```

```
Thread #0 is alive: true
Thread #1 is alive: true
Thread #2 is alive: true
Thread #3 is alive: true
T1 thread exiting.
T1 thread exiting.
T2 thread exiting.
T2 thread exiting.
Thread #0 is alive: false
Thread #1 is alive: false
Thread #2 is alive: false
Thread #3 is alive: false
Main thread exiting.
```

CS370 Worksheet 8 Name_____

1) What is the difference between preemptive scheduling and non-preemptive scheduling? (P. 186)

2) What is the role of the dispatcher in an OS? What is dispatch latency? (P. 187)

3) What are the criteria a scheduler uses? What is each? (P. 187)

4) Name 2 scheduling algorithms. (P. 189)