

Setting name and priority.

```
Main thread is now called: Thread #1
Priority is now: 8
```

5. You need to be careful about what operations you perform on the main thread. For example, if you add the following code to the end of `main()`, the program will never terminate because it will be waiting for the main thread to end!

```
try {
    thrd.join();
} catch (InterruptedException exc) {
    System.out.println("Interrupted");
}
```



Module 11 Mastery Check

1. Why does Java's multithreading capability enable you to write more efficient programs?
2. Multithreading is supported by the _____ class and the _____ interface.
3. When creating a runnable object, why might you want to extend **Thread** rather than implement **Runnable**?
4. Show how to use `join()` to wait for a thread object called **MyThrd** to end.
5. Show how to set a thread called **MyThrd** to three levels above normal priority.
6. What is the effect of adding the **synchronized** keyword to a method?
7. The `wait()` and `notify()` methods are used to perform _____.
8. Change the **TickTock** class so that it actually keeps time. That is, have each tick take one half second, and each tock take one half second. Thus, each tick-tock will take one second. (Don't worry about the time it takes to switch tasks, etc.)
9. Why can't you use `suspend()`, `resume()`, and `stop()` for new programs?
10. What method defined by **Thread** obtains the name of a thread?
11. What does `isAlive()` return?
12. On your own, try adding synchronization to the **Queuc** class developed in previous modules so that it is safe for multithreaded use.