**Chapter 11:**

Self-Test:

* *How does Java’s multithreading capability enable you to write more efficient programs?* Use resources more often rather than having them do nothing.
* *Multithreading is supported by the \_\_\_\_\_ class and the \_\_\_\_\_\_\_ interface.* Thread class and Runnable interface.
* *When creating a runnable object, why might you want to extend Thread rather than implement Runnable?* Use it to override the Thread class methods.
* *Show how to use join() to wait for a thread object called MyThrd to end.* MyThrd.join();
* *Show how to set a thread called MyThrd to three levels above normal priority.* MyThrd.setPriority(Thread.NORM\_PRIORITY + 3);
* *What is the effect of adding the synchronized keyword to a method?* Only one thread can use the method at a time.
* *The wait() and notify() methods are used to perform \_\_\_\_\_\_\_\_\_\_\_?* Interthread communication.
* *Change the TickTock class so that is actually keeps time.* Implemented in TickTock.java
* *Why can’t you use suspend(), resume() and stop() for new programs?* They’re deprecated due to causing bad issues.
* *What method defined by Thread obtains the name of a thread? getName()*
* *What does isAlive return? True if thread is running, False if terminated*
* *On your own, try adding synchronization to the Queue class developed in previous chapters so that it is safe for multithreading use.* Can’t find the file so this can’t be done.