**J6 When is it necessary to use a Thread-Safe collection in a Java program?**

Multithreading, a feature of Java that allows for the simultaneous operation of several threads, is well known. We will receive inconsistent results if many threads are working on the same set of data while the value of that data is changing since that situation is not thread safe. Thread-safety is the procedure used when one thread is actively working on an item while preventing another thread from doing the same. If the library function is not "thread-safe," you may still offer atomic access to it by using a lock in your code. Some thread-safe, effective implementations of collections, such as linked lists and hash tables, are provided by the java.util.concurrent package. Through the application of complex algorithms, thread-safe collections reduce contention by enabling concurrent access to various data structure components.

**J7 If a Thread-Safe collection is needed, but not available, what are your other options?**

A class that is thread-safe ensures that its internal state and method returns are accurate even when they are called simultaneously from different threads. Java's thread-safe collection classes include Stack, Vector, Properties, Hashtable, etc. If a Thread-Safe collection is not available but I need to use it in my program, then I will used lock and synchronized to achieve the thread-safe.