Multi-threaded programs are susceptible to a program called a Race condition. A Race condition happens when a block of code is executed by more than one threads simultaneously, and program execution no longer follows a predictable path. In this section, I am going to show you how to write a multithreaded program by creating and starting threads, and then we will deal with the Race condition by using a special technique called thread locking.

But thread locking is only part of the solution. For two threads to safely exchange data, we also need some kind of mechanism to synchronize the threads because let’s face it, the first thread needs to pick up the data after the second thread has finished producing it and not the other way around. I will conclude the section by teaching you a technique to synchronize two or more threads.

By the end of the section, you will have a solid grasp of multithreaded code techniques and you’re going to need this knowledge when we start working with Tasks and Parallel LINQ queries.