1. 
2. **Problem**: Synchronized collection achieves thread safety by using lock. So a thread gets access to synchronized collection and other threads have to wait. It works in most of the cases but it can have negative impact on **performance & scalability** as no of threads (concurrent) operations increases.  
   **Solution**: Concurrent collections in java. These classes use “**Partition Technique”** to allow concurrency. So, they divide different data into segment and different threads work on those segments. Only one thread at a time can access that segment.   
   **Therefore**, Concurrent Collections are faster than Synchronized Collections because they don’t use synchronization.  
   **package** 🡺 java.util.concurrent
3. 