**CopyOnWriteArrayList** **implements the List interface** like ArrayList, Vector, and LinkedList but **it's a thread-safe collection**.

As the name suggests **CopyOnWriteArrayList** **creates a copy of underlying ArrayList with every mutation operation** e.g. add or set. Normally CopyOnWriteArrayList is **very expensive** because it **involves costly Array copy with every writes operation** but it's **very efficient** if you have a List **where Iteration outnumbers mutation** e.g. we mostly need to iterate the ArrayList and don't modify it too often.

**Iterator** of **CopyOnWriteArrayList** is **fail-safe** and **doesn't throw ConcurrentModificationException** even if **underlying CopyOnWriteArrayList is modified once Iteration begins** because **Iterator is operating on a separate copy of ArrayList**. Consequently, all the updates made on CopyOnWriteArrayList are not available to Iterator.