COPYONWRITEARRAYLIST

- DEFINATION OF CONPYONWRITEARRAYLIST ?

  "Copy On Write " means that whenever a write operation like adding or removing element instead of directly modifying the existing list , a new copy of the list is created and the modification is applied to that copy this ensure that others thread reading the list while it's being modified are unaffected.

  Read Operations: Fast and Direct, since they happen on a stable list without interference from modifications.

  Write Operations: A new copy of the list is created for every modification after modification the reference to the list is then updated so that subsequent(next reads) reads use this new list.

- CODE FOR COPYONWRITEARRAYLIST:

```
public class CopyOnWriteArrayList_01 {

    public static void main(String[] args) {

/* will throw ConcurrentModificationException
        List<String> shoppingList = new ArrayList<>();

        shoppingList.add("Milk");
        shoppingList.add("Eggs");
        shoppingList.add("Bread");
        System.out.println("Initial Shopping List: "+shoppingList);

        for (String item : shoppingList) {
            System.out.println(item);
            if(item.equals("Eggs")){
                shoppingList.add("Butter");
                System.out.println("Added Butter While Reading.");
            }
        }
        System.out.println("updated shopping list: "+shoppingList);
 */


        // this will work fine

        List<String> shoppingList = new CopyOnWriteArrayList<>();

        shoppingList.add("Milk");
        shoppingList.add("Eggs");
        shoppingList.add("Bread");
        System.out.println("Initial Shopping List: "+shoppingList);

        for (String item : shoppingList) {
```

```
                System.out.println(item);
                if(item.equals("Eggs")){
                    shoppingList.add("Butter");
                    System.out.println("Added Butter While Reading.");
                }
            }
            System.out.println("updated shopping list: "+shoppingList);

        }
    }
```