HASHSET

(1) Internal working of hashset?

- internally uses hashmap to store elements (Similar to HashMap)
- Only contains unique elements
- Only key it takes and dummy value
- It is unordered
- Backed by HashMap (keys are stored, values are dummy)

Ex-

```
public static void main(String[] args) {

    // Create a HashSet
    Set<Integer> set = new HashSet<>();

    // Add elements
    set.add(3);
    set.add(5);
    set.add(7);
    set.add(3); // Duplicate, will not be added

    // Display the HashSet
    System.out.println("HashSet: " + set);

    // Check if an element is present
    System.out.println("Contains 5? " + set.contains(5));

    // Remove an element
    set.remove(5);
    System.out.println("After removing 5: " + set);

    // Get the size of the HashSet
    System.out.println("Size: " + set.size());

    // Check if the HashSet is empty
    System.out.println("Is Empty? " + set.isEmpty());

    // Use iterator to traverse
    System.out.println("Traversing HashSet:");
    for (Integer i : set) {
        System.out.println(i);
    }

    // Clear the HashSet
    set.clear();
    System.out.println("After clear: " + set);

    // For thread safety
    Set<Integer> set1 = Collections.synchronizedSet(new HashSet<>(set));
```

```
    ConcurrentSkipListSet<Integer> set2 = new ConcurrentSkipListSet<>();

    // For synchronized traversal, use synchronized block
    synchronized (set1) {
        for (Integer i : set1) {
            System.out.println(i);
        }
    }
}
```