- Can we use HashMap to sort the data using comparable or comparator?

No, a hash map has no sense of ordering. However, you can easily sort the keys or values with a one-liner:

SortedSet<String> keys = new TreeSet<String>(myHashMap.keySet());

- We have a remove method in the ArrayList, we have remove method in the iterator? Why the duplicity of these two methods?

We don’t want to have only the iterator remove method because we would have to iterate to that point to begin the removal, instead of O(1) access. (The operation is still O(n) technically.)

We need the Iterator remove method because it is implemented in other lists, like LinkedList where there is no performance loss iterating to the middle of the list (since there is no random access).

The iterator remove method may also have performance advantages if you have to iterate down the list no matter what.

- Internal Working of HashSet

A HashSet is implemented with a HashMap, and each element added into the HashSet is a new key in the HashMap, with a constant value (so the value has no meaning).

More detail: A hashing function is run on any element being added to the HashSet. That hash result is used to index into a backing array, allowing O(1) random access speed.