(7) HashSet in Java.

1) add()
2) remove()
3) contains()

} O(1) — All return true when successful

4) HashSet<non-primitive type> h = new HashSet<String>();
(Integer, Student, etc)

5) System.out.println(h); → [ele1 ele2 ele3]

6) Iterator<String> i = h.iterator();
while(i.hasNext()){
System.out.println(i.next()+" ");
}
→ ele1 ~~ggg~~ ~~ele2~~ ele3

7) Order can be anything while accessing.

8) Instead of iterator.
for(String s: h)
{ System.out.println(s+" "); }

9) size()
remove()
isEmpty()
clear()
} Extra fns.

HashMap in Java

(1) • Hashset only stores keys, Hashmap stores key:value pair.

(2) ⇒ `HashMap<String, Integer> m = new Hashmap<String, Integer>();`

(3) ⇒ `m.put("gfg", 10);` { Other fns: m.containsKey() & containsValue(). m.remove() → takes key, returns value removed.

(4) • Traversal ⇒
```
for (Map.Entry<String, Integer> e : m.entrySet())
{
    System.out.println(e.getKey() + " " + getValue());
}
```

(5) `m.get("key/value");` → or you can specify like above print..
↳ if nothing there null returned.