



HashMap vs HashSet

This lesson will discuss the difference between HashMap and HashSet and look at some of their key features.

We'll cover the following



- Introduction
- 🔍 HashMap?
- 🔍 HashSet?

Introduction

Before we solve any challenges regarding Hash Tables, it is necessary to look at the definitions of **HashMap** and **HashSet** and see how they are different. Both are implemented in Java using the **Hash Table** class, which is why it is also a common misconception that these two structures are the same, but they are very different from each other.

🔍 HashMap?

HashMap is a collection that contains all the key-value pairs; it *maps* the values to keys. There is a built-in class available in Java for HashMap, implemented by using **Map** interface. It provides the basic functionality of hashing along with some helper functions that help in the process of insertion, deletion, and search.

Some of the key features of HashMap are given below:



A HashMap stores key-value pairs (examples given below) to map a key to the value:


$$abc \rightarrow 123$$
$$xyz \rightarrow 456$$

- HashMap cannot contain duplicate keys. It can, however, have duplicate values.
- HashMap also allows multiple **null** values and only one **null** key
- This mechanism does **not** support synchronous operations and is **not** thread-safe.



HashSet?

HashSet class is implemented in Java using **Set** interface. It is also built in the same way as HashMap, i.e., using the Hash Table class, but it is still quite different from the HashMap class. Some of the key features of HashSet are listed below:

- HashSet also stores values in an unordered way, using hashing, but this happens in the backend. On the backend, the HashSet class is implemented using the HashMap class. The value that we add to the HashSet is then added to the HashMap as a key, corresponding to a dummy value Object. The retrieval remains $O(1)$
- HashSet is a class which implements the Set interface and this interface only stores values, not a key-value pair.
- HashSet restricts storing multiple **null** values and only allows **one null** value in the whole table
- HashSet does not allow storing duplicate values as a **set** can only contain unique elements
- Just like HashMap, HashSet also does **not** support synchronous operations



Enough of concepts for now. It's time to test your understanding. You gotta solve some mind crunching exercises in the upcoming lessons. See you there!

Interviewing soon? We've partnered with Hired so that companies apply to you
[utm_source=educative&utm_medium=lesson&utm_location=US&utm_campaign=educative](https://hired.com/?utm_source=educative&utm_medium=lesson&utm_location=US&utm_campaign=educative)



← Back

Next →

Trie vs Hash Table

Challenge 1: Find whether an array is ...



Mark as Completed



Report an Issue



Ask a Question

(https://discuss.educative.io/tag/hashmap-vs-hashset__hash-tables__data-structures-for-coding-interviews-in-java)