

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

**Smart Programming**
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies
City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : <http://www.smartprogramming.in>

BUY COURSES ON : <https://courses.smartprogramming.in>

 YOUTUBE CHANNEL : Smart Programming (<https://www.youtube.com/c/SmartProgramming>)

 ANDROID APP : Smart Programming
(<https://play.google.com/store/apps/details?id=com.smartprogramming>)

 <https://www.facebook.com/smartprogramming.india>
 https://www.instagram.com/smart_programming



Hashtable & Working of Hashtable & HashMap in Java

=> Hashtable :-

- ➔ Hashtable is a direct implemented class of Map interface which is present in java.util package
- ➔ Syntax : public class Hashtable extends Dictionary implements Map, Cloneable, Serializable { - }
- ➔ Hashtable was introduced in JDK 1.0 version
- ➔ Hashtable is also known as legacy class
- ➔ The underline data structure of Hashtable is "Hashtable"

➔ Properties of Hashtable :-

1. Hashtable stores the values in key-value pair and each key-value pair is known as entry
2. In Hashtable, keys should always unique but values can be duplicate
3. Hashtable can store heterogeneous elements at key position
4. In Hashtable we cannot insert null values at key or value position

5. Hashtable does not follows the insertion order by default
6. Hashtable does not follows the sorting order by default
7. Hashtable is synchronized map
8. Hashtable does not allows more than one thread at one time
9. Hashtable allows the sequential execution
10. Hashtable increases the execution time which in turn makes our application slow
11. Hashtable is thread-safe
12. Hashtable provides the guarantee for data consistency

➔ Constructors :-

1. `public Hashtable() { - }` //it will create a Hashtable having 11 as capacity and default fill ratio is .75%
2. `public Hashtable(Map t) { - }`
3. `public Hashtable(int initialCapacity) { - }`

4. public Hashtable(int initialCapacity, float loadFactor) { - }

→ Methods :-

= same methods as that of Map interface

→ When we should use Hashtable :-

= Hashtable is good for searching or retrieval operation

=> Working of Hashtable :-

→ "hashCode" is the unique integer value of each and every object that is provided by JVM

→ Hashtable initialCapacity is 11

- ➔ Then for each and every key hashcode value will be generated and its index position will be calculated by using hashing technique
- ➔ And at that index position that key-value pair or entry will be inserted
- ➔ If two elements have same index position, then that entry will be inserted at right side of previous entry
- ➔ When the values are traversed then they are traversed from top to bottom and right to left

Hashtable hm=new Hashtable();

-> it will create a hashtable of 11 capacity
-> every part is known as bucket

where 101 is hashcode and 11 is default capacity of hashtable

ht.put(101, "deepak"); $101 \% 11 = 2$
ht.put(105, "kamal"); $105 \% 11 = 6$
ht.put(104, "deepesh"); $104 \% 11 = 5$
ht.put(102, "rahul"); $102 \% 11 = 3$
ht.put(106, "amit"); $106 \% 11 = 7$

hm.put(116, "aaa"); $116 \% 11 = 6$

System.out.println(hm);

(from top to bottom and from right to left)

{106=amit, 116=aaa, 105=kamal, 104=deepesh, 102=rahul, 101=deepak}

	10
	9
	8
106 : amit	7
105 : kamal 116 : aaa	6
104 : deepesh	5
	4
102 : rahul	3
101 : deepak	2
	1
	0

1. Hashing technique : it is a technique by which we convert each hascode value into indices

2. Hash-collision : it is that situation when 2 keys will have same indices

=> Working of HashMap :-

- ➔ HashMap initialCapacity is 16
- ➔ For every entry it will calculate the index position and store the element there.
- ➔ If multiple entries have same index position, then it will create linked list and starts storing in that linked-list
- ➔ In java 8 updates, after threshold value is filled then it starts storing the elements in balanced tree



=> Difference between HashMap & Hashtable :-

1. HashMap was introduced in 1.2 version
Hashtable was introduced in 1.0 version
 2. HashMap is not a legacy class
Hashtable is a legacy class
 3. In HashMap we can store the null values
In Hashtable we cannot store the null value at key or value position
 4. HashMap is non-synchronized Map because
HashMap does not contain any synchronized methods
Hashtable is synchronized Map because it contains synchronized methods
 5. Multiple points related to synchronization
-

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: <https://www.smartprogramming.in/>
<https://courses.smartprogramming.in>

Android App:
<https://play.google.com/store/apps/details?id=com.smartprogramming>

YouTube Channel:
<https://www.youtube.com/c/SmartProgramming>