| **Hash map** | **Hash table** |
|---|---|
| Not legacy class<br>Not synchronized / thread safe<br>Performance is high<br>Can add 1 null key and many null value | Legacy class<br>synchronized / thread safe<br>Performance is low<br>Null key and value not allowed |

**Map**

- Map is not child interface of collection
- If we want to represent group of object as key-value pairs then we should use map
  Ex. 101-Amit, 102-Vaibhav
- Both keys and value are objects only
- Duplicate keys are not allowed, but value can be duplicated
- Each key-value pair is called **entry** hence map is consider as collection of entry objects

Map interface methods

- put(key, value) //add key value pair to map, if key already present then old value will be replaced with new value and return old value
- putAll(map m) //add group
- get (key) // returned value associated with specified key
- remove (key) //remove the entry associated with specified key
- containsKey()
- containsValue()
- isEmpty()
- size()
- clear()
- keySet() // return key set
- values() // return values set
- entrySet()
- getKey()
- getValue()
- setValue()

Properties:

- underlying data structure – Hash table
- insertion order is not preserved, and based on hash code of keys
- heterogenous objects are allowed for both keys and values
- null key is allowed only once
- null value is allowed (any number of times)
- best choice for frequent search operation

| **Hashmap** | **HashTable** |
|---|---|
| Not synchronized | synchronized |
| Not thread safe | thread safe |
| High performance | Low performance |
| Null key and value allowed | Not allowed |
| Not legacy | Legacy |

**Program:**

```
package CollectiionProg;

import java.util.HashMap;
import java.util.Iterator;
import java.util.Set;

public class HashMapProg {

        public static void main(String[] args) {

                HashMap hm = new HashMap();

                hm.put(3, "Ferrari");
                hm.put(2, "Lamborgini");
                hm.put(1, "Mastang");
                hm.put(4, "Porsh");

                System.out.println(hm);
                hm.put(null, "Twitter");
                System.out.println(hm);
                hm.put(null, "Meta");
                System.out.println(hm);
                hm.put(5, "ABC");
                hm.put(6, "ABC");
                hm.put(10.5, 'a');
                System.out.println("HM = " + hm);

                System.out.println("------------");
                System.out.println(hm.keySet());
                System.out.println(hm.values());
                System.out.println(hm.entrySet());

                System.out.println("------------");

                Set a = hm.entrySet();

                Iterator itr = a.iterator();

                System.out.println("For Output ------------");
                for(;itr.hasNext();)
                {
                        System.out.println(itr.next());  //8    = 8
                }

                System.out.println("While Output ------------");
                while(itr.hasNext())
                {
                        System.out.println(itr.next());
                }
```

```
	}
}
```

### 1) Child Browser

We can inspect, drag and drop, contain add, max, min and close option

Set (string) ids = driver.getWindowHandles();  
String w1 = ids.get(1);  
driver.switchTo().window(w1);

getWindowHandles() => Return all window address (main and child)  
To handle the child browser popup we need to switch selenium focus from main page to child browser by using following syntax:  
driver.switchTo.window(String abc);

**Program:**

Link1 :- https://www.aspsnippets.com/demos/1102/  
Link1 :- https://skpatro.github.io/demo/links/

```
package Popup;

import java.util.Iterator;
import java.util.Set;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Windows1 {

	public static void main(String[] args) throws Exception {
		System.setProperty("webdriver.chrome.driver", "chromedriver");
		WebDriver driver = new ChromeDriver();
		driver.manage().window().maximize();
		driver.get("https://www.aspsnippets.com/demos/1102/");

		driver.findElement(By.xpath("//input[@value='Open Popup']")).click();

		Set<String> windows = driver.getWindowHandles();

		Iterator itr = windows.iterator();
//      System.out.println("All windows");
//      while(itr.hasNext())
//          System.out.println(itr.next());
        System.out.println("No of elements in collection = " + windows.size()); //2

        String win[] = new String[3];
        for(int i=0;i<windows.size();i++)
        {
            win[i] = (String) itr.next();
            System.out.println(win[i]);
        }

        driver.switchTo().window(win[0]);  //Main window
        System.out.println("Title of Win 0 = " + driver.getTitle());
        driver.close();
        Thread.sleep(5000);
        driver.switchTo().window(win[1]);  //Small window
        System.out.println("Title of Win 1 = " + driver.getTitle());
        driver.close();


//      Thread.sleep(5000);
//      driver.close();
    }

}
```