

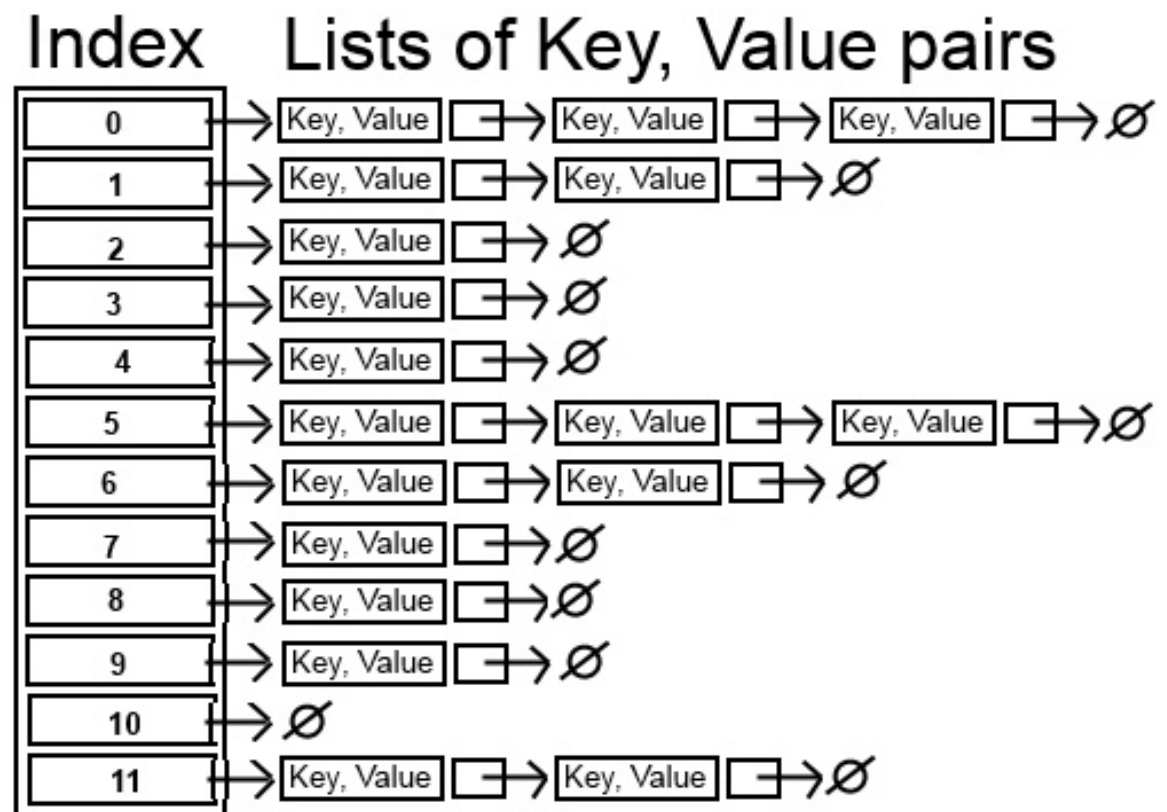
Hashtable

- › Hashtable collection contains a group of elements of key/value pairs stored at respective indexes.

› Full Path:
System.Collections.Hashtable

› Process of adding an element:

1. Generate index based on the key. Ex: $\text{index} = \text{hash code} \% \text{size of hashtable}$
2. Add the element (key and value) next to the linked list at the generated index.



'Hashtable' collection

```
Hashtable referenceVariable = new Hashtable();
```



- › It is dynamically sized. You can add, remove elements (key/value pairs) at any time.
- › Key can't be null or duplicate; but value can be null or duplicate.
- › The "Hashtable" class is not a generic class.
- › You can set / get the value based on the key.
- › It is not index-based. You need to access elements by using key.

Properties

- › **Count** : Returns count of elements.
- › **[TKey]** : Returns value based on specified key.
- › **Keys** : Returns a collection of key (without values).
- › **Values** : Returns a collection of values (without keys).

Methods

- › **void Add(object key, object value)** : Adds an element (key/value pair).
- › **void Remove(object key)** : Removes an element based on specified key.
- › **bool ContainsKey(object key)** : Determines whether the specified key exists.
- › **bool ContainsValue(object value)** : Determines whether the specified value exists.
- › **void Clear()** : Removes all elements.