

The Hashtable class represents a collection of **key-and-value pairs** that are organized based on the hash code of the key. It uses the key to access the elements in the collection.

A hashtable is used when you need to access elements by using **key**, and you can identify a useful key value. Each item in the hashtable has a key/value pair. The key is used to access the items in the collection.

Properties and Methods of the Hashtable Class

The following table lists some of the commonly used **properties** of the **Hashtable** class:

Property	Description
Count	Gets the number of key-and-value pairs contained in the Hashtable.
IsFixedSize	Gets a value indicating whether the Hashtable has a fixed size.
IsReadOnly	Gets a value indicating whether the Hashtable is read-only.
Item	Gets or sets the value associated with the specified key.
Keys	Gets an ICollection containing the keys in the Hashtable.
Values	Gets an ICollection containing the values in the Hashtable.

The following table lists some of the commonly used **methods** of the **Hashtable** class:

S.N	Method Name & Purpose
1	Public Overridable Sub Add <i>keyAsObject, valueAsObject</i> Adds an element with the specified key and value into the Hashtable.
2	Public Overridable Sub Clear Removes all elements from the Hashtable.
3	Public Overridable Function ContainsKey <i>keyAsObject</i> As Boolean Determines whether the Hashtable contains a specific key.
4	Public Overridable Function ContainsValue <i>valueAsObject</i> As Boolean Determines whether the Hashtable contains a specific value.
5	Public Overridable Sub Remove <i>keyAsObject</i> Removes the element with the specified key from the Hashtable.

Example:

The following example demonstrates the concept:

```
Module collections
  Sub Main()
    Dim ht As Hashtable = New Hashtable()
    Dim k As String
    ht.Add("001", "Zara Ali")
    ht.Add("002", "Abida Rehman")
    ht.Add("003", "Joe Holzner")
    ht.Add("004", "Mausam Benazir Nur")
    ht.Add("005", "M. Amlan")
    ht.Add("006", "M. Arif")
    ht.Add("007", "Ritesh Saikia")
    If (ht.ContainsValue("Nuha Ali")) Then
      Console.WriteLine("This student name is already in the list")
    Else
      ht.Add("008", "Nuha Ali")
    End If
    ' Get a collection of the keys.
    Dim key As ICollection = ht.Keys
    For Each k In key
      Console.WriteLine(" {0} : {1}", k, ht(k))
    Next k
    Console.ReadKey()
  End Sub
End Module
```

When the above code is compiled and executed, it produces the following result:

```
006: M. Arif
007: Ritesh Saikia
008: Nuha Ali
003: Joe Holzner
002: Abida Rehman
004: Mausam Banazir Nur
001: Zara Ali
005: M. Amlan
```

Loading [Mathjax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js