Collections : HashTable

- C# Hashtable class is a collection class where we can store data in the key/value pair.
- Hashtable is not type-safe.
- Hashtable is not generic and both key and value are based on object type.
- Hashtable is used to store unique keys data.
- If we try to assign value to already used key, it just overrides the previous stored data.
- It is based on the Hashing algorithm which is optimized for retrieving the data fast.
- It only takes O(1) operation to retrieve any value based on key.
- Hashing algorithm is based on hash function.
- A hash function is any function which maps variable size key data to a particular fixed size data.

Property of HashTable class

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.

Methods of HashTable

public virtual void Add(object key, object value);
Adds an element with the specified key and value into the Hashtable.

public virtual void Clear();
Removes all elements from the Hashtable.

public virtual bool ContainsKey(object key);
Determines whether the Hashtable contains a specific key.

public virtual bool ContainsValue(object value);
Determines whether the Hashtable contains a specific value.

public virtual void Remove(object key);
Removes the element with the specified key from the Hashtable.

Application Program which demonstrate use of HashTable methods and properties.

```
using System;
using System.Collections;

public class Marvellous
{
	public static void Main(string[] args)
	{
		Hashtable ht = new Hashtable();

		ht.Add("10","Pre-Placement Activity");
		ht.Add("11","Angular");
		ht.Add("12","Logic Building");
		ht.Add("13","Inndustrial Project Development");

		ICollection keys = ht.Keys;

		Console.WriteLine("Elemnts of Hashtable after removal");

		foreach (String k in keys)
		{
			Console.WriteLine(ht[k]);
		}

		ht.Remove("11");

		Console.WriteLine("Elemnts of Hashtable after removal");

		foreach (String k in keys)
		{
			Console.WriteLine(ht[k]);
		}
	}
}
```