C# - Collections

- The **System.Collections** namespace contains interfaces and classes that define various collections of objects, such as lists, queues, bit arrays, hash tables and dictionaries.
- C# also includes specialized classes that hold many values or objects in a specific series, that are called 'collection'.
- Collection classes are specialized classes for data storage and retrieval. These classes provide support for stacks, queues, lists, and hash tables. Most collection classes implement the same interfaces.
- Collection classes serve various purposes, such as allocating memory dynamically to elements and accessing a list of items on the basis of an index etc. These classes create collections of objects of the Object class, which is the base class for all data types in C#.
- There are two types of collections available in C#:

1.Non-generic	2.Generic
ArrayList	List
HashTable	Dictionary
SortedList	SortedList
Stack	Stack
Queue	Queue

- **In non-generic collections**, each element can represent a value of a different type. The collection size is not fixed. Items from the collection can be added or removed at runtime.
- **Generic Collections** work on the specific type that is specified in the program whereas non-generic collections work on the object type.

1. HashTable:

- A hash table is used when you need to access elements by using key, and you can identify a useful key value.

- ❖ Each item in the hash table has a **key/value** pair. The key is used to access the items in the collection.
- ❖ Hashtable stores key and value pairs. It retrieves the values by comparing the hash value of the keys.

2. **Sorted List:**
   - ❖ It uses a **key** as well as an **index** to access the items in a list.
   - ❖ A sorted list is a combination of an array and a hash table.
   - ❖ If you access items using an index, it is an ArrayList, and if you access items using a key, it is a Hashtable.
   - ❖ It automatically arranges elements in ascending order of key by default.
   - ❖ C# includes both, generic and non-generic SortedList collection.
3. **Stack:**
   - ❖ Stack stores the values in LIFO style (Last In First Out).
   - ❖ It provides a **Push()** method to add a value and **Pop() & Peek()** methods to retrieve values.
   - ❖ C# includes both, generic and non-generic Stack.
4. **Queue:**
   - ❖ Queue stores the values in FIFO style (First In First Out).
   - ❖ It keeps the order in which the values were added. It provides an **Enqueue()** method to add values and a **Dequeue()** method to retrieve values from the collection.
   - ❖ C# includes generic and non-generic Queue.
5. **BitArray:**
   - ❖ It represents an array of the **binary representation** using the values 1 and 0.
   - ❖ BitArray manages a compact array of bit values, which are represented as Booleans, where true indicates that the bit is on (1) and false indicates the bit is off (0).