## Collections

- A Collection is a group of objects
- .NET Framework contains a large number of classes and interfaces that define and implement collections
- collections are mainly use for data storage and retrieval
- collection are two types
  - non -generic collections
  - generic collections

## Collections

- Non-Generic Collections:
  - In Non Generic Collections  the data stores  in the form of object.
  - Non Generic Collection classes and Interfaces  are defined in the **System. Collections**
  - **In  System. Collections** classes  stores any type of information in the form of a object and returns in object type
  - when we perform any mathematical operations it requires type casting.
  - they are not type -safe

## Collections

- Non -Generic Collection Classes are
- ArrayList
- Hashtable
- SortedList
- Stack
- Queue

## Collections

- System. Collections Defines a number of non-generic interfaces, these interfaces are determine the functionality common to all of the non generic collection classes
  - ICollection:
    - Defines elements that all non generic collections must have.
    - It is the base Interface for all non generic collection classes.
    - Members:
    - Count:it is a property which return that number of items held in the collection.
    - CopyTo():it copies the contents of a collection to the array

# Collections

- IList:
  - The IList Interface declares the behavior of a non generic collection that allow elements to accessed via a zero based index.
  - Implements **`ICollection`**, **`IEnumerable`**
  - Methods
  - Int Add(object obj):Add the object into invoking collection
  - Void Clear():Delete all the elements from invoking collection
  - Bool Contains(object obj)-it determines whether the given object is contains in the invoking collection list or not
  - Void Insert(int idx,object obj):Insert obj at the index specified by idx
  - Void Remove(object obj):Removing the object from invoking collection
  - Void RemoveAt(int idx):Remove the object at the index specified by idx

# Collections

- IDictionary:
  - IDictionary stores the objects in the from of key/value pairs.
  - Once the pair is stored, you can retrieve it by using its key.
  - IDictionary implements ICollection and IEnumarable
  - Methods
    - Void Add(object k,object v):Adds the key/value pair to the invoking collection.
    - Void Clear():Removes all key/value pairs from the invoking collection
    - Bool contains(object k):Returns true when invoking collection contains given key value.
    - Remove(object k):Removes the entry whose key equals to k

## Collections

- IComparer:
  - The IComparer interface defines a method compare(),which defines that two objects are compared in collection.
  - int compare(object v1,object v2)
- IEnumarable: Defined the GetEnumarator() method which supplies the enumerator for a collection class
- IEnumarator: Provides methods that enables the contents of a collection to be obtained one at a time.
- IDictionaryEnumaration:Defines the enumerator for a collection that implements IDictionary.

# Collections

- Generic Collections:
  - Generic Collection Classes are introduce from .NET 2.0 onwards
  - Generic Collections classes and Interfaces are defined in **System.Collections.Generic**
  - Generic Collection Classes provides
    - increased type-safety
    - provide better performance

## Collections

- Generic collection classes are:
  - List<type>
  - Dictionary class<key, value>
  - Sorted Dictionary<key, value>
  - SortedList<key,value>
  - stack<type>
  - queue<type>
  - LinkedList<type>
- Generic collection Interfaces are:
- ICollection<T>
  IComparer<T>
- IDictionary<Tk, Tv>
- IEnumarable<T>
- IEnumarator<T>
- IList<T>