## Objective

* To use non – generic and generic classes provided in collection framework.
* To implement in-built collection interfaces.
* To create a custom generic class.

## Assignments to be done in this session

1. Create following types of arrays
   1. Integer
   2. String

Use System.Array class to perform following operations on them

Copy, Sort, Clear, Reverse

Accept input from user through Console.

1. Use collection class such as ArrayList to hold more than one employee objects in Employee Management application. Display all Employee details which are stored in collection.
2. Write a console based program to create a linked list of Employee objects using the generic class List<>.Perform following operations on the list:
3. Add a new employee
4. Display the list of employees.
5. Total number of employees in the list
6. Write Custom Generic class MyStack based on assignment of previous session, with

Push() and Pop() methods to store any kind of .NET Type.

## Now try these to get a complete grip…

1. In the assignment 3 above, add a functionality to search an employee on name in the List<>.
2. Create a class named Player that contains Player name, runs scored as data members. Create a class named Team that contains an array of Player. Implement IEnumerable interface for class Team.

Write a console based application to create an object named India. Use foreach loop to iterate through the object India to display information about its players.

1. Use an iterator to iterate through the players in the above example.

## By the way

* Generic classes offer better performance over non-generic counterparts as issues like casting and boxing are reduced.
* yield keyword is used iterate through a collection. This enhances the performance of an application.
* Custom Collection classes can be created by implementing the Collection Interfaces provided by .NET framework class library.