Day 28 26/03/2025

generic collections :-

using System . collections . Generic ;

List <int> lt = New List <int> ();

$$\begin{tikzcd}
\text{List <int>} \arrow[d] & \text{New List <int> ()} \arrow[d] \\
\text{Datatype} & \text{Datatype}
\end{tikzcd}$$

lt . Add ( ) ↓

* we can onlyadd value of which data type we given .

* it similar to Array list .

dictionary <int, string> dr = New dictionary <int, string> () ;

$$\begin{tikzcd}
\text{int} \arrow[d] & \text{string} \arrow[d] \\
\text{key} \arrow[d] & \text{value.} \\
\text{Two data type.} &
\end{tikzcd}$$

* same as hashtable only allow unique key .

dr . Add (1, "Hello");

For (key value pair< int, string> item in dr)

To get both value.

* we can only add value and key on which ~~type~~ datatype we given.

<u>Stack</u> :- Same as stack

stack < int > s = New stack <int>();

<u>Queue</u> :- same as queue.

Queue <int> s = new queue <int>();

s ~~orted~~ dictionary <int, string > dr1 =
new sorted dictionary <int, string>

same as dictionary and Sorted dictionary will store in sorted order.

SortedList <int, string> SL = New SortedList<int, String> ();

↓
Same as sorted dictionary but SortedList is fast.

Hashset -> unorder

Hashset <int> HS = New Hashset <int> ();

↓
unique

eg:- student S.no ← rollnumber
↓
unique

HS.Add (1); → we can not ~~[illegible]~~ add 1 ~~[illegible]~~ again.
HS.Add (2);

Sortedset -> ordered

sortedset <int> SS = New sorted set <int> ();

↓
same as Hashset except sortedset is sorted (ordered).

In generic collection we can use custom type.

eg :- List<Student> L;

```
public class Student {

    Public int Rollno;
    Public String name;
    public String email;

}

Public void ..... {
  if (L is null)
  {
    L = new List <student>();
  }
  Student std = new Student();
  Std. Rollno (7);
  std. name ("Name");
  std. email ("email.gmail.com");
  L. add (std);
}
```