

Yesterday:-

1. constructor level dependency injection
→ why interfaces.

2. Properties

3. wrapper method → insert, update, delete
+ logger

4. Events & Delegate

+ = coupling operator

- = de-coupling operator

+ = → multicast delegate.

5. Object Pooling → static

Today :-

1. Collection F/W, Generics
 2. File IO & Serialization →
 3. .dll → access modifier.
-

✓ `int [] arr = new int [5];`
 ↳ Array `arr = new Array();`



`arr[0] = 1;`

→ `arr.Add(1);`

`[1|2|3|4|5]`

string

str

↳ `String str = new String();`

`char[]`

Non-Generic
collection

Iterable ✓

↓
ICollection

→ IDictionary

↓
Hashtable

↘ IList

↓
class ArrayList

↓ search by index

Single value and
index based
collection

↓
Key-Value pair
based collection
→ search by
"key"

class Array

Compulsory
you have to
mention its
Datatype.

int [] arr;

Obj [] arr
 ↓ managed heap

10 | 11 | 23.33 | emp | "par" | book

(item is Book)
 Book book = item
 as Book;

BN, BN

id | Name | Ad

if (item is Emp)

{ Emp empref = item as Emp;

empref.Id.

Generic Collection

$IEnumerable<>$



$ICollection<>$

$ICollection<>$

$IDictionary<TKey, TValue>$



$class List<>$

$class Dictionary<key, value>$

[Growable
+ Fixed datatype
+ index based]

[Growable
+ Fixed datatype for
key and value
+ key based search]

Library files \rightarrow .dll

\rightarrow reusable code.

interface I Database
 {
 }

 IDb.dll

