Yesterday :—

1) Non-generic collection :— IEnumerable
   int[], obj[], ArrayList, Hashtable, stack, Queue
2) Generic collection :— IEnumerable<>
   List<>, Dictionary<>, foreach loop → use
3) Generic classes and Generic method.
4) Generic overloadded delegates
5) Tuples collection
6) ranges and indices
7) Dynamic Datatype :— initialize at runtime.
8) out parameter
9) Pass by value & Pass by reference.

Today:-

1] File IO & Serialization

2] Reflection

Finitely Infinite

Stream :— data in a channel -: bits and bytes.

System.IO;

abstract Class Stream

Persist
↳ Data on the Hard drive

Class FileStream ✓

Class NetworkStream
Persist and transfers data across the network [TCP etc]

class CryptoStream
For encrypting data and transferring

Class FileStream

5th / 6th

FileStream fs = new FileStream ("FilePath"

"C:/myApp/Data.txt",

FileMode.CreateOROpen,

Enum

FileAccess.Write);

StreamWriter writer = new StreamWriter(fs)

writer. .Write() → one line

.WriteLine() → one l...

writer.Flush()

writer.Close() → fs.Close();

For Lab :-

Hotel App :- Console App. / logic should be in dll file

Finitely Infinite

↳ Serice Online

User → Show menues option [Veg, Non-Veg, Both]

User → Veg → Display list of items like
Name of Dish, Price

Same for Non-Veg →

Both → Combine menus
→ Name of Dish, Price

User → select multiple menues.

→ Display msg :— C.W ("order is on the way")

→ File.txt → Order No :-
list of ordered items & total bill.

<? xml version="1.0" ?>

xml schema

<Emp>

<Id> 10 </Id>

<Name> Mugdha </Name>

<Address> Pune </Address>

</Emp>

typeof (Emp)

↑

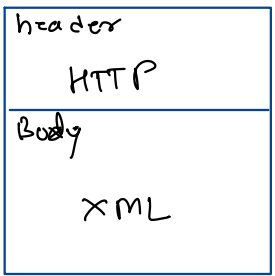
at the time of initialization

SOAP formatter

[Simple Object Access Protocol]

SOAP Protocol = HTTP + XML ;

HTTP Packet



```
public class Cmath
{
    p. int Add (x,y)
    {
        return int;
    }
    p. int Sub (x,y)
    {
        return (int);
    }
}
```

Cmath.dll

Compile

```
Assembly asm = new Assembly();
asm.type = types;
Type[] types = new Type[2]{"Cmath", "Book"};
```

```
Type type = new Type()
type.Name = "Cmath.Cmath";
type.isPublic = true;
type.isClass = true;
type.isInterface = false;
type.isSealed = false;
type.isInherited = false;
type.methods = allmethods[];
MethodInfo[] methods = allmethods[];
method[0].Name = "Add";
method[0].isPublic = true;
method[0].isOverriden = false;
method[0].ReturnType = "System.Int32";
method[1].Name = "Sub";
method.Parameters = new ParameterInfo[2];
parameter.Name = "x";
parameter.ParameterType = "System.Int32";
parameter.Count = 2;
```

menu . dll

class menu

{

int i

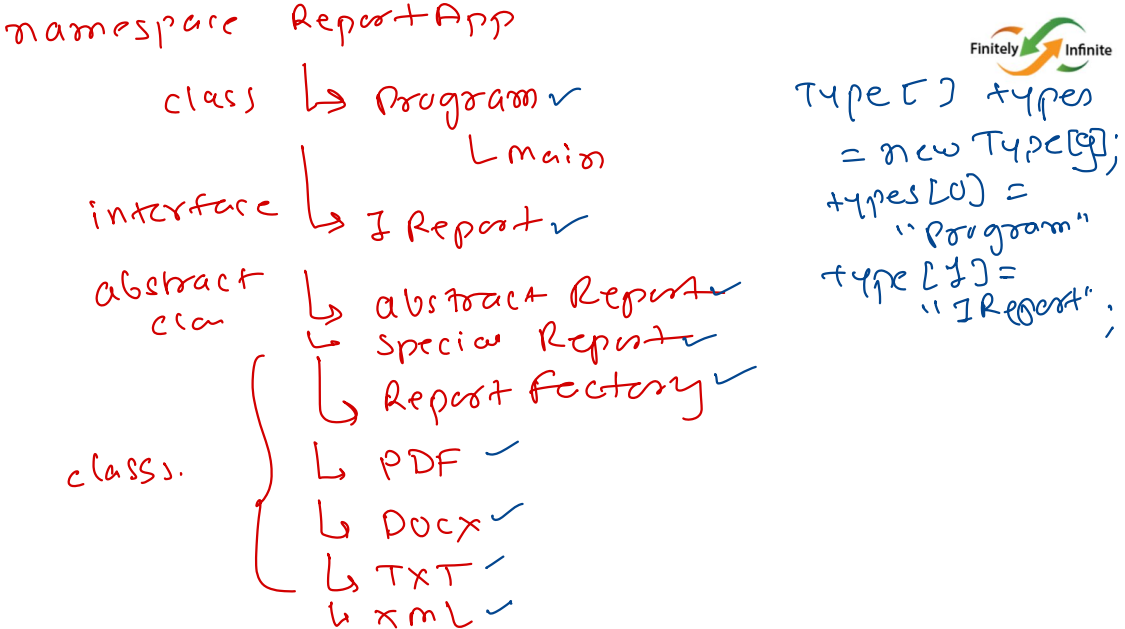
prop List<vegmenu> vegmenues;

↳ = new List<vegmenu>();

[0] = new vegmenue();

↳ obj. itemName = "";

}


namespace ReportApp
class
Program
Main
interface
IReport
abstract class
abstractReport
SpecialReport
ReportFactory
classes.
PDF
Docx
TXT
xml
Type[] types = new Type[9];
types[0] = "Program"
type[1] = "IReport";
Finitely Infinite