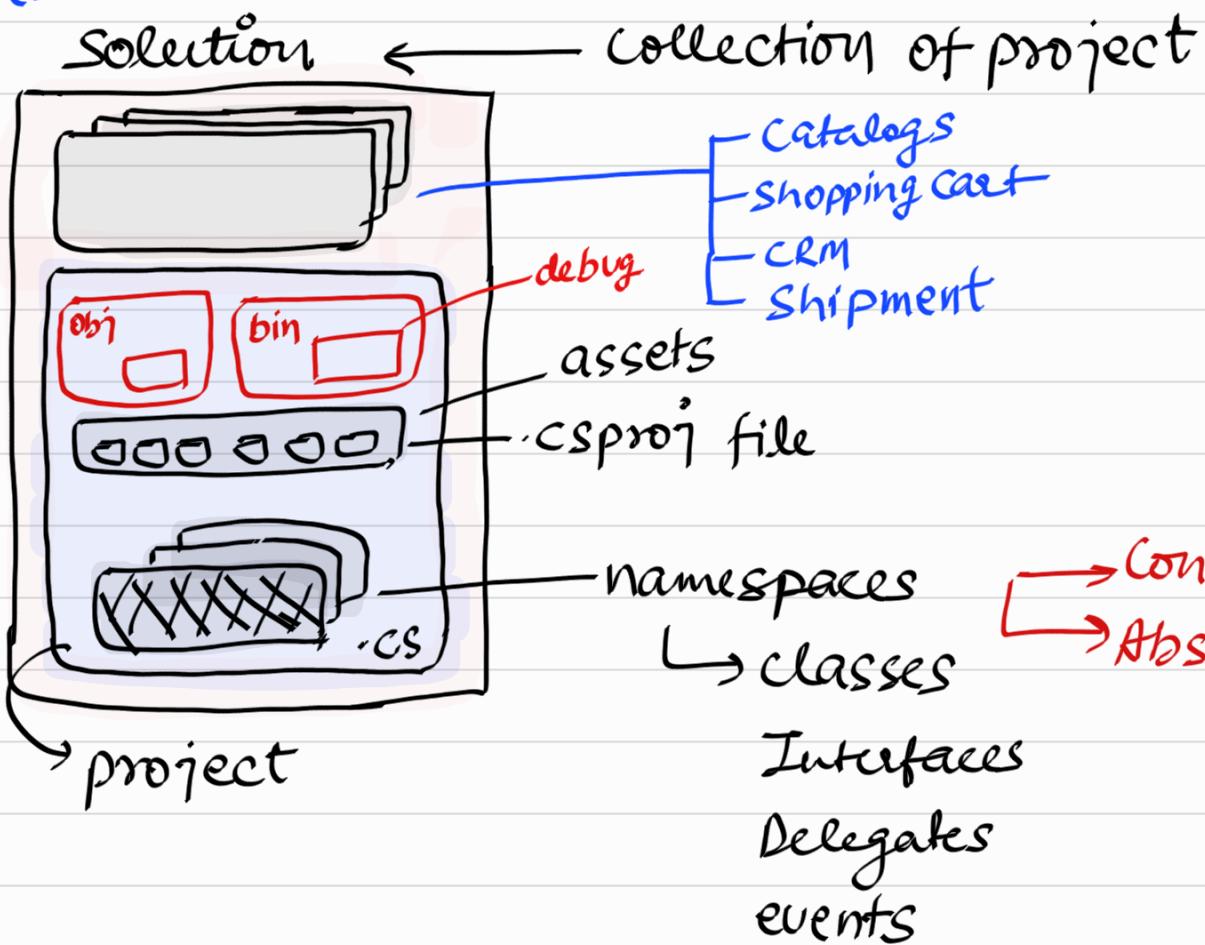
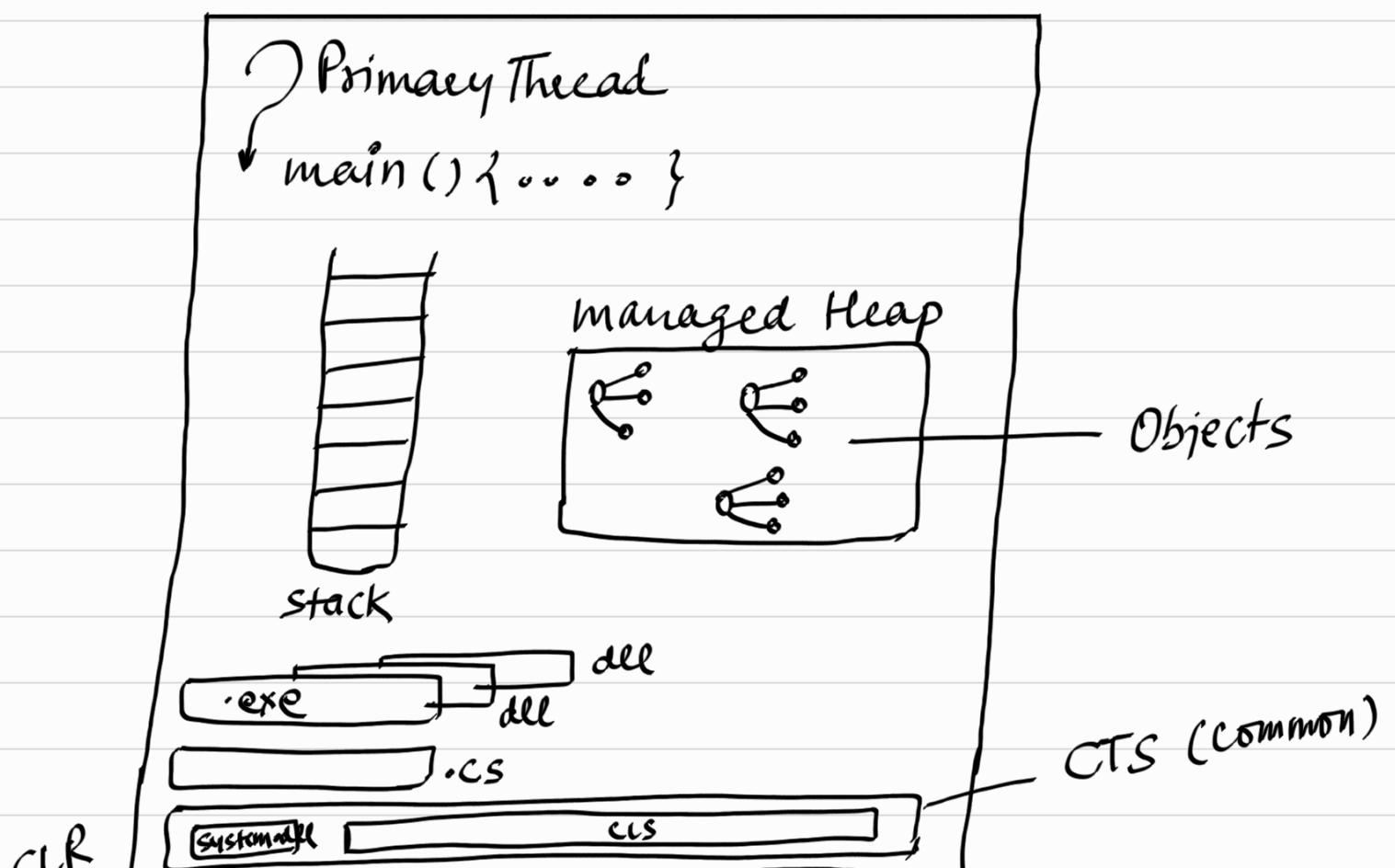
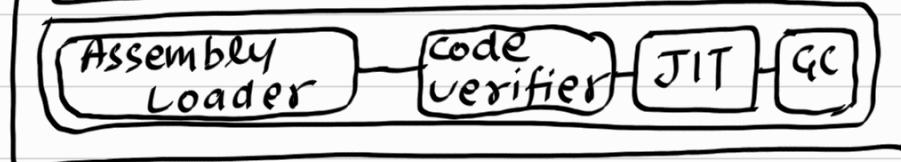


(Ecommerce Solution)



abstract class is used to achieve polymorphism.





Mark and sweep Method

- used by Garbage Collector
- in OOPS.

Create folder Ecommerce

Inside folder

Cmd → dotnet new sln

for console application — This has main
and Runs.

dotnet new console -o Test

for class library

dotnet new classlib -o membership

All classlibs now should be linked with
Ecommerce solution.

dotnet CLI commands check

dotnet sln add membership\membership.csproj

when all class libraries are added then

Start coding.

Setters and Getters Shortcut :-

```
private string firstname;  
  
public string FirstName {  
    get { return firstname; }  
    set { firstname = value; }
```

```
public string LastName { get; set; }
```

Shortcut
for
getter
setter

Called autoproperty

under membership namespace

- ① class Autmanager
- ② class User

```
Console.WriteLine ("Enter name");
```

```
String name = Console.ReadLine();
```

(for Reading i/P

To use a property from one project to Another project use command.

first go to target folder the write command.

Test

dotnet add reference ..\Membership\membership.csproj

|| This will include property of membership in Test project.

|| To check go to Test.csproj file and see.

|| will look like

address

<ItemGroup>



<ProjectReference Include="..\member..." />

</ItemGroup>

/>

To Build

dotnet build (being the solution)

To Run

Go to folder where program.cs is
dotnet run

□ Like java in dotnet there is Object class which is mother of all classes.
i.e. superclass of all classes.

members of Object class

ToString

GetHashCode

Equals

GetType \Rightarrow GetClass in java

In java we used super keyword here
here we use base.

As Override annotation in java
here

public override string ToString()
{

}

Constructor Chaining :-

default constructor

:this()

parameterised Constructor

Employee

//Constructor Chaining

public Employee():this(12,"Ravi", "Tambade", "ravi.tambade@transflower.in", "9881735801",

"Pune", new DateTime(1975, 8, 18), 1000, 10000, 15000){

}

↓
(Chaining)

virtual:- (not Compulsive overriding)

If we want to override the method from
Other class.

It may or may not be overridden.

(It may or may not be overridden).

- public virtual float ComputePay() { };
- public abstract void Work();
↳ Compulsive Overriding

~~//~~ public SalesManager (int id, string firstname, string lastname)
:base (id, firstname, lastname);
↳ Sending data to parent class.

In Employee-HR Code

If we want to use HR and membership
in Test then

eCommerce .SLN

{

Console file - main

Class Libraries - classlib

