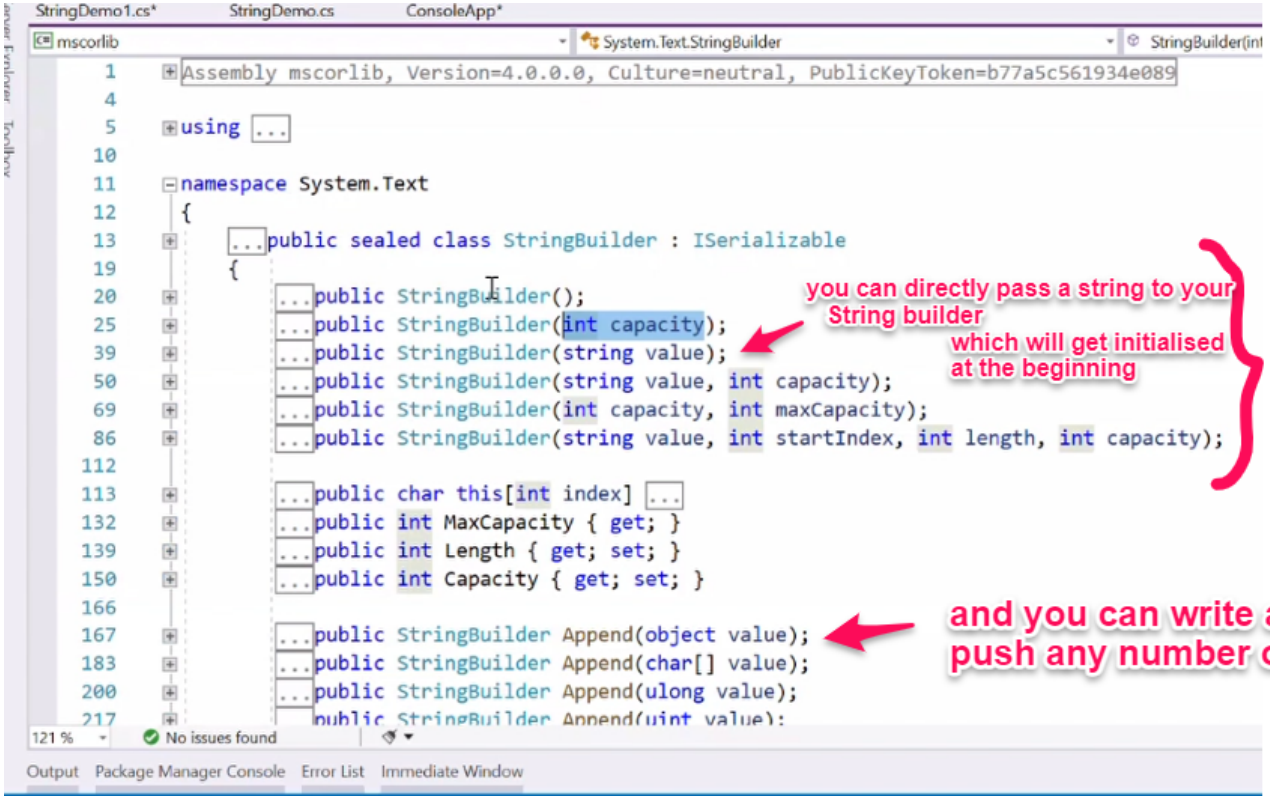
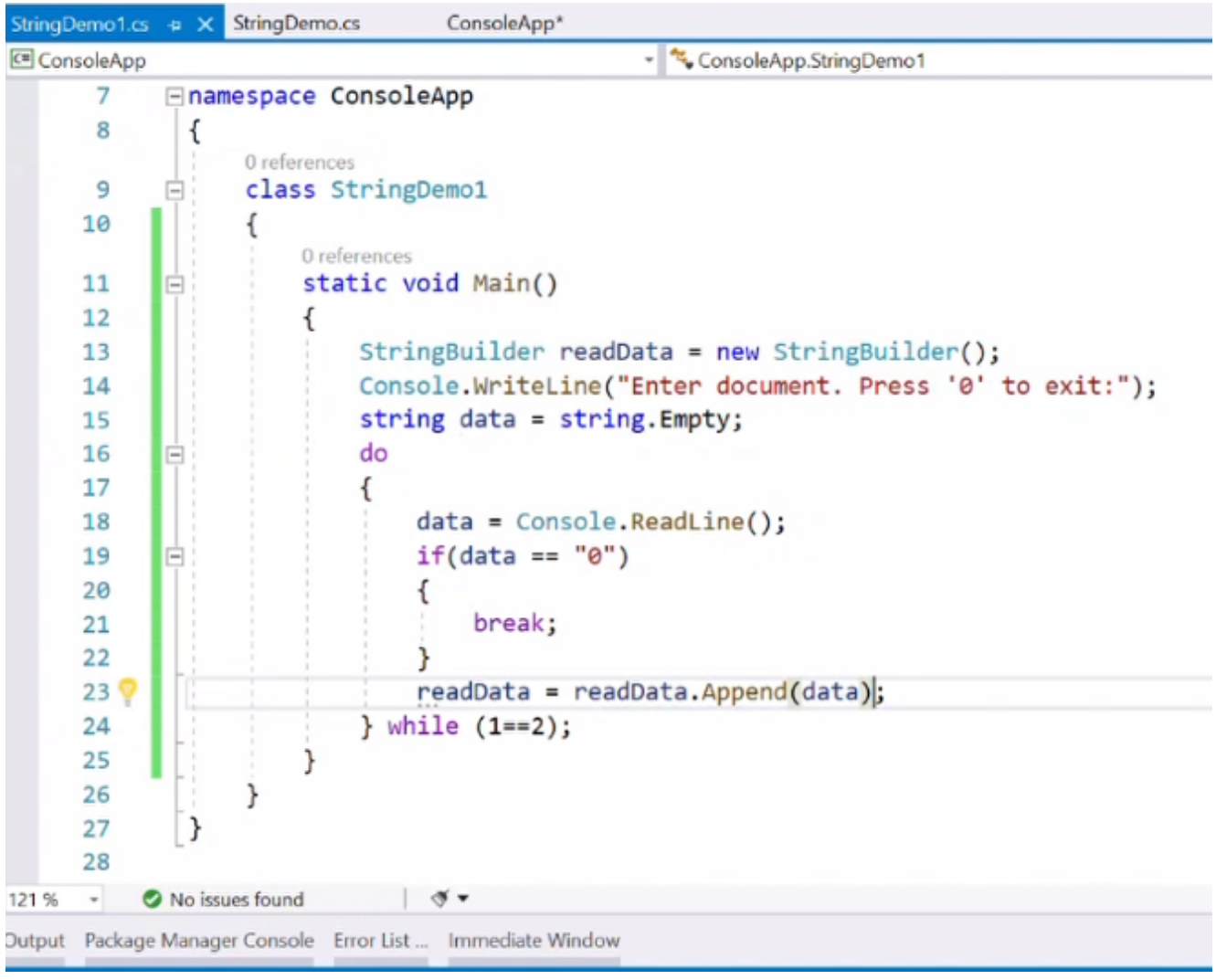
C#-6



you can directly pass a string to your String builder which will get initialised at the beginning

you can see it has 6 overloads

and you can write a uppend command and you can push any number of character or object


StringDemo1.cs
StringDemo.cs
ConsoleApp*
ConsoleApp
ConsoleApp.StringDemo1
namespace ConsoleApp
{
0 references
class StringDemo1
{
0 references
static void Main()
{
StringBuilder readData = new StringBuilder();
Console.WriteLine("Enter document. Press '0' to exit:");
string data = string.Empty;
do
{
data = Console.ReadLine();
if(data == "0")
{
break;
}
readData = readData.Append(data);
} while (1==2);
}
}
}
121 %
No issues found
Output
Package Manager Console
Error List ...
Immediate Window

If i say i want to have a particular newspaper it should always give me the updated information but it should not be a paper, may be you can imagine a particular file which will have some kind of paper kind of thing and based on action whatever user performs if will move to the next page previous page, so i want to create some thing like this so all those things becomes object, Now there is just an idea

So that is called as a **Class**

Now what class should have ?

It will say what my object should have

Class

And what my object will have, **Object will have its own state and behaviour**

What is the state of an object ?

--state of the object is the value what you have

for example name, each one of us have our own name so name identifies me, so name is the state and what we have data inside that, that becomes the value for it

Age, all of us have age, all of us have date of birth, so all those things becomes a state

What is the Behaviour?

Behaviour is the ways how it works

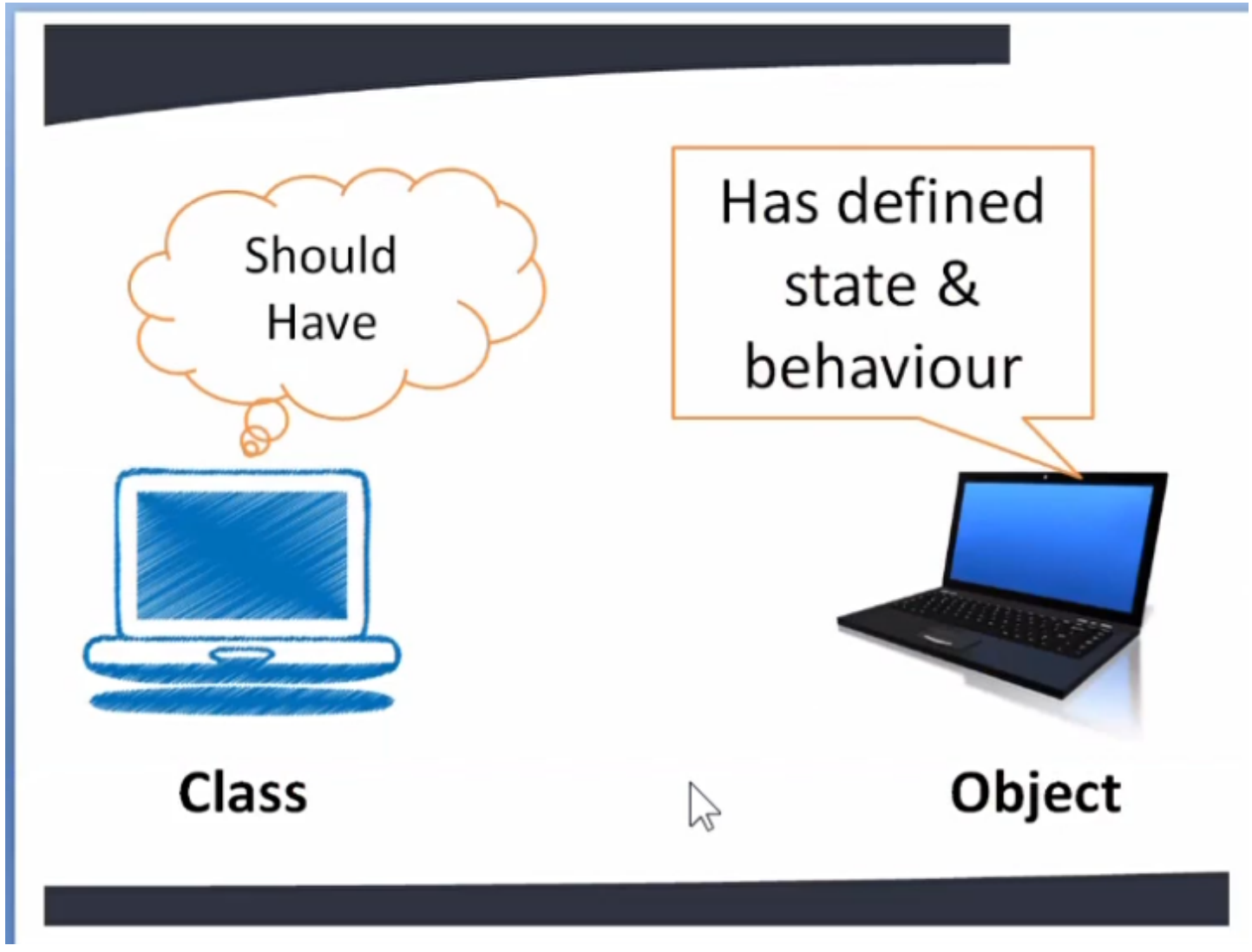
What is a character?

Is Character and Behaviour both are same?

if somebody comes and say your behaviour is not good What will you do ?-----u'll feel bad

if somebody comes and say your character is not good What will you do ?-----we'll first slap them then we'll listen tell me now why do you say that my character is not good

Because when you say somebodies character is not good, it represents your parents it comes from the genes, it is something which you cannot change, character is something which makes you
But what about behaviour, that is something which can be changed
Behaviour is a reaction to a stemulous, it may be like a mosquito comming and biting you so some people will kill that mosquito, some will say keep drinking🤣 i don't mind, drink and go but don't make sound
But character is something which makes u so character cannot be changed



How do objects interact with each other ?

Laptop has a motherboard

Laptop has a CD-Rom

Laptop has a Key Board

So this is **has a** Relationship

Now how do you relate a laptop and a television?

they both have LED they both fall under the electronic devices

So, i can say laptop is a electronic gadget

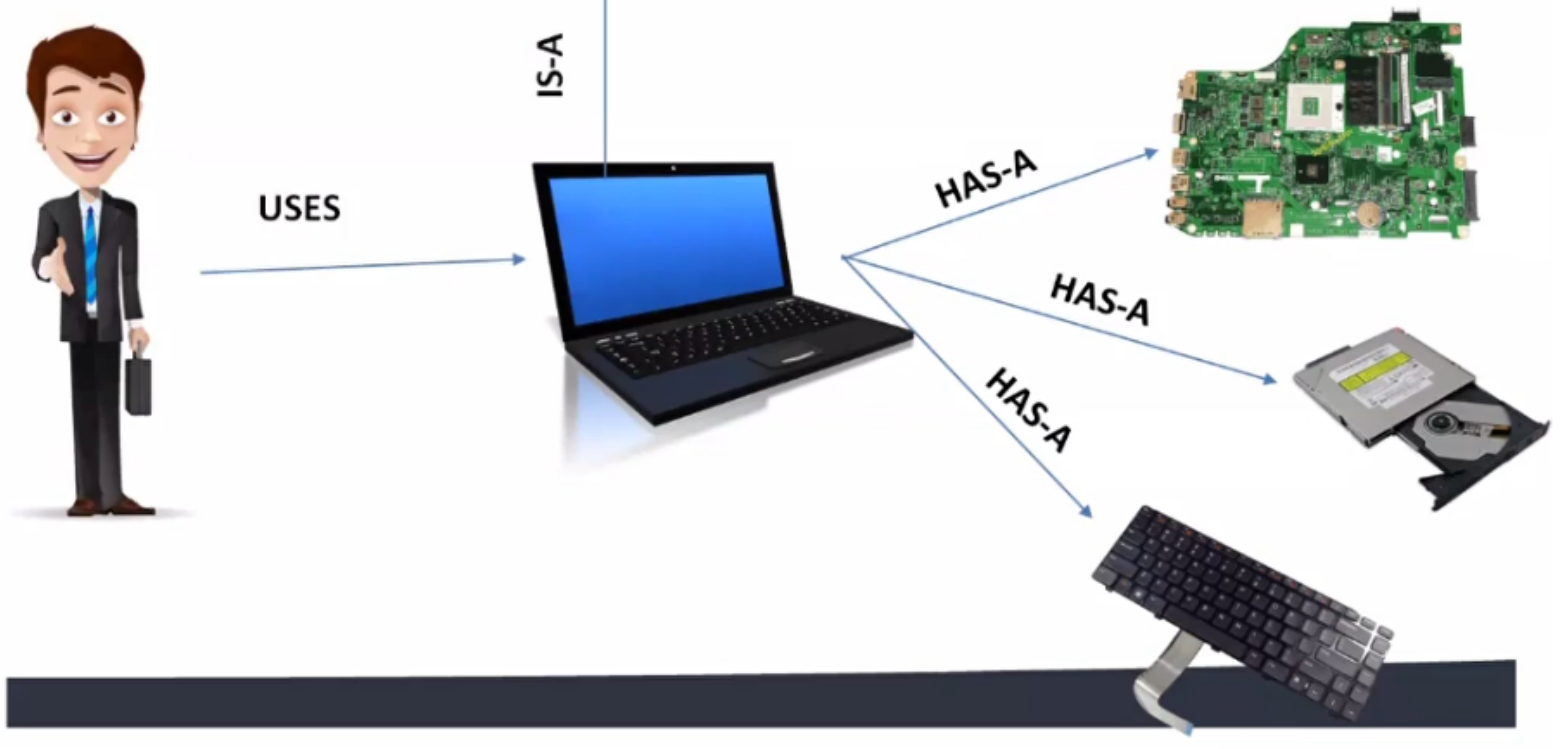
and we use those particular gadgets

So, we have three different types of relationships

1. **Is a**
2. **Has a**
3. **Uses**

Interaction of the Objects

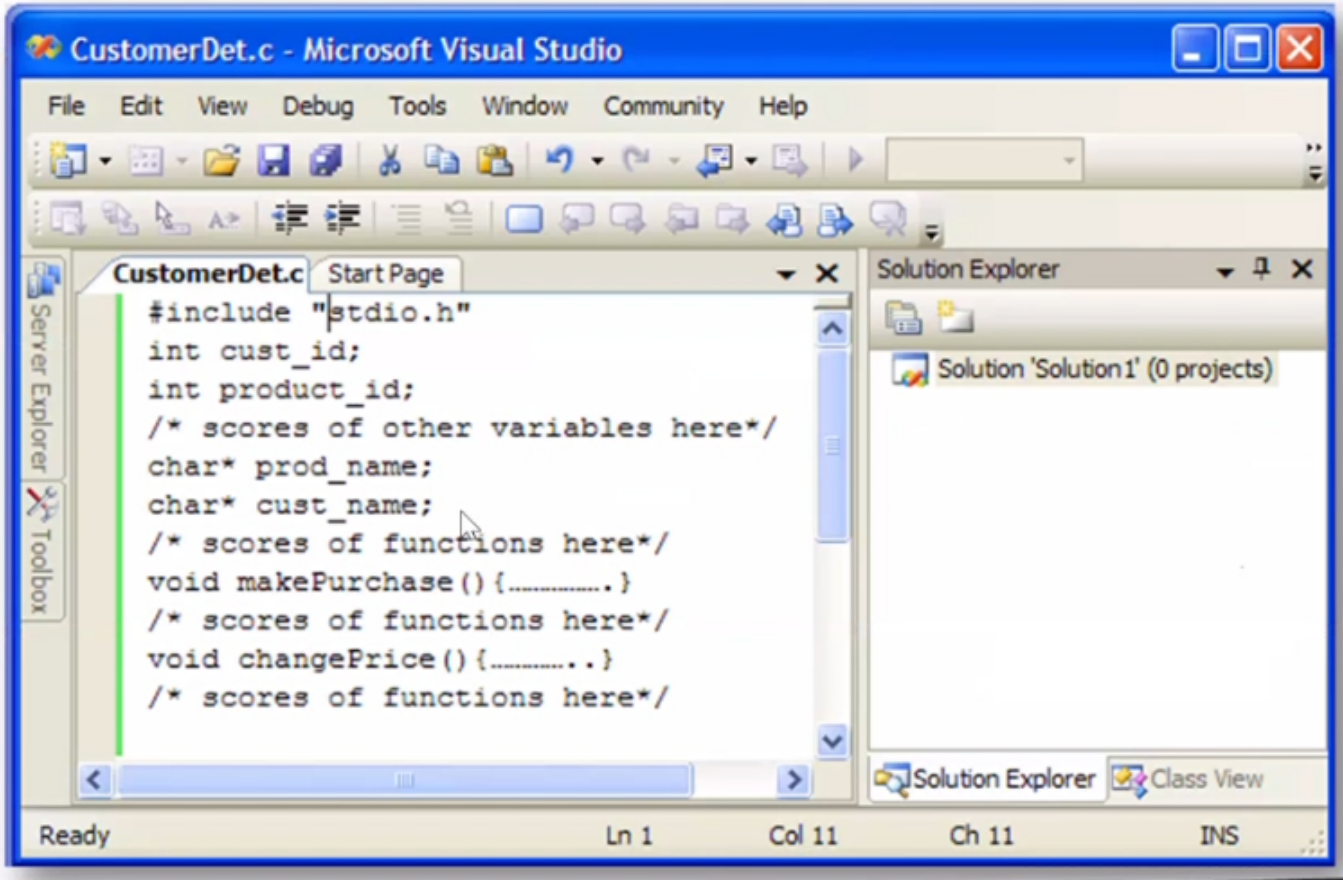
Electronic Gauged



Why object oriented?

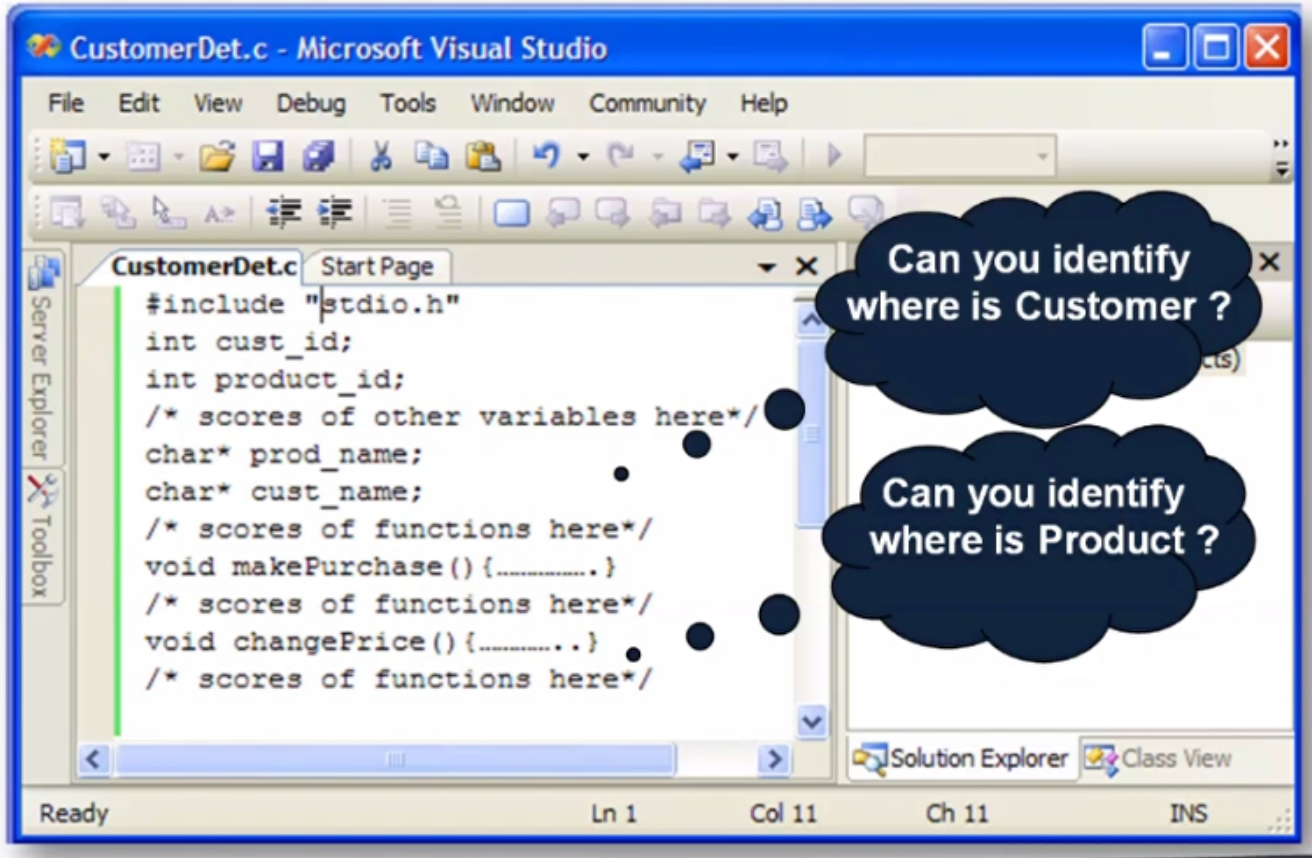
identifying and differentiating becomes easy in object oriented

Why choose the OO approach?



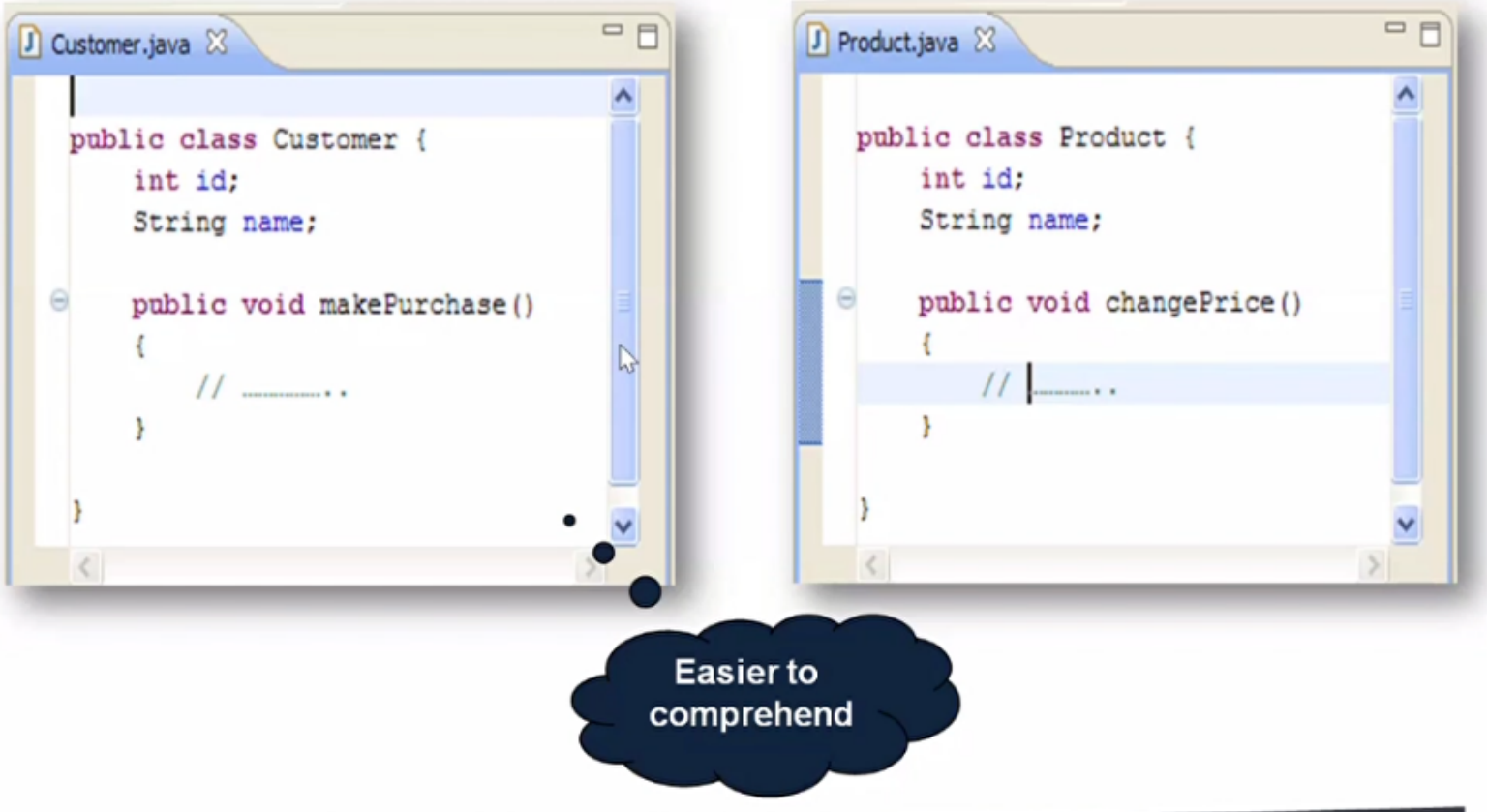
Look here when ever you writing a C program So, it will put every thing together like i have a product information, i have a customer information.....
it puts every thing together, now as the project grows up so identifying and seprating these particular things becomes very difficult, maintenance becomes very difficult, debugging becomes very difficult because every thing is present in the same .c file

Why choose the OO approach?



So, i object oriented we'll keep all the customer related information in customer class and product related information in product class
so if tomorrow if i want to add some information in customer so i'll just go to the customer class without touching the product class i can make modifications
that's how maintenance becomes easy

Why choose the OO approach ?



Now when you talk about objects we have three different types of object

1. Physical Entity :- something which is physically available something which i can touch and feel like employee, library, customer
2. Conceptual Entity :- which we cannot touch and feel like account, order, training, policy
3. Software Entity :- which my software has like list, linked list.some kind of data structure what we have

Objects & its types

- We interact with *objects* everyday
  - A customer
  - An order
  - Your car
  - Your Mobile

- An object represents an entity – physical, conceptual or software
  - **Physical entity**
    - Employee, Customer, Supplier
  - **Conceptual entity**
    - Account, Policy, FeesCalculator
  - **Software entity**
    - List, Connection, etc.
- ***A programmer should make a good effort to capture the conceptual entities in addition to physical entities which are relatively straight forward to identify***

Why choose object oriented?
object oriented is always talk about classes
In procedure oriented world its the functions which becomes a building block but when you talk about object oriented it's a classes which becomes the building block and in object oriented we speak about the real world
meaning i have a bank, i can create a bank in software, i have a student i can create a student in my program
So, this becomes a real world modelling
So, when you talk about object oriented we group the code with the behaviour
we group the data to the methods associated to it
And it will allow me to work with the abstractions

What is Abstraction?

in final year you had your abstract report correct, which describes about your project, So while giving abstract about your project, will you say ok this is something which is hidden or you'll say that this is something which i'm showing you

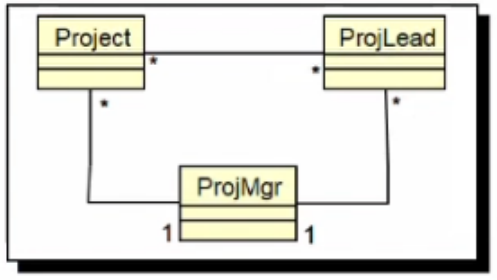
When you holding a remote, so are you bothered about what is hidden to or you bothered about what is visible to you, so when i am holding a remote i'm bothered more about what is visible to me which i can see, when i press this button what should happen i should know this,
when you driving a car you know that when i press the accelerator it will speed up that is something which i know, but am i intrested that how it works what happened internally----------No

So Abstraction --No doubt that it hides the information but while telling don't say like you hide the information,

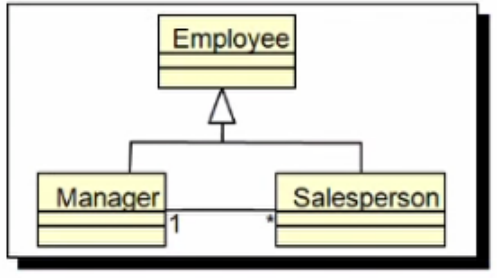
Abstraction is something which is shown to you, which you can see

Why choose the OO approach?

- The OO approach
  - Deals with classes as the building blocks
  - Allows Real World Modeling
  - The idea of OOP is to try to approach programming in a more natural way by grouping all the code that belongs to a particular object—such as an account or a customer — together



- Raise the level of abstraction
  - Applications can be implemented in the same terms in which they are described by users
- Easier to find nouns and construct a system centered around the nouns than actions in isolation



- Easier to visualize an encapsulated representation of data and responsibilities of entities present in the domain
- The modern methodologies recommend the object-oriented approach even for applications developed in C or Cobol

Now How do you identify the particular objects -----its very simple you have to always identify the nouns

Classes will encapsulate the member data and member methods, so data will always be hidden so don't allow anybody to access this directly and if i want to access them, so it should be only through member methods

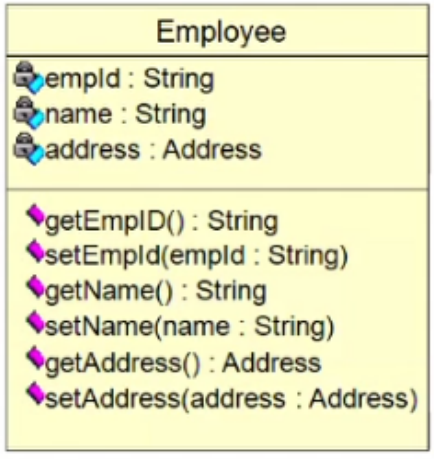
So, i'll gone to encapsulate my member data and member methods

Encapsulating means i'll make them as one group, i'll make sure that they will always be present together , so they cannot be seprated i cannot pull them out it becomes one unit

So, class will have single representation for your data members and member methods

Class

- User defined type
  - Encapsulates all the data and operations pertaining to an entity
  - Provides a Single representation to all the attributes defining the entity
  - Passing single representations is easier



- Data types as collections
  - A struct in C encapsulates only data. Used as a data structure to store different types of data
  - An array is used to store different elements of the same type

So, UML is mainly used to design the entire system what ever you are using
So in UML when you talk about **is a** we have **Generalization and realization**

when you talk about **Has a** we have **association, aggregation, composition**
when you talk about **Uses** we have **Dependency**

Relationship between Classes

- **Classification**

<<Is-a>>	<<Has-a>>	<<Uses>>
Generalization Realization	Association Aggregation Composition	Dependency

- For all practical purposes we will represent
  - Is-a relationship as 
  - Has-a relationship as
  - Uses relationship as 

Has a relationship has 3 most important things

1. **Multiplicity** means how many object one object will contain

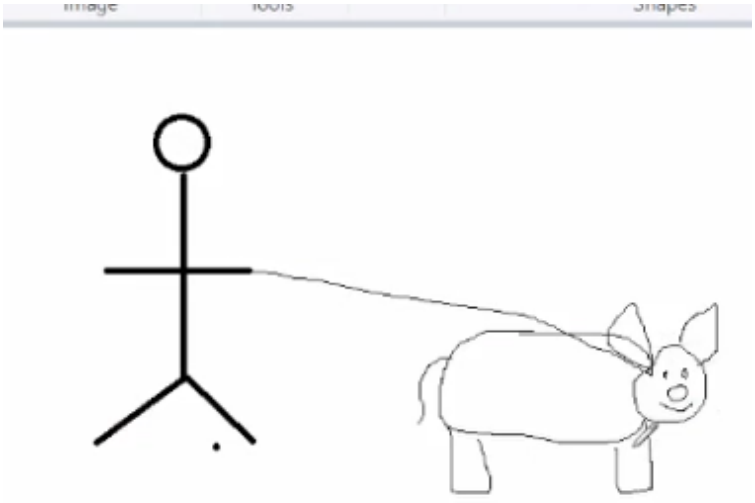
   for example How many refill a pen will have

   1 pen will always have one refill

   1 pen may have 4 refills

   so that is multiplicity how many object the main object has in it

2. **Navigation** means Who owns this particular object



Suppose we have a person who owns a dog, and he travel with the dog, he carries the dog whenever he for evening walk or something like that

So here man owns dog

Whenever in the object oriented when ever you draw this you will always put a line like this

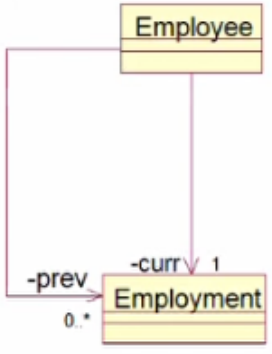
this says one way relationship

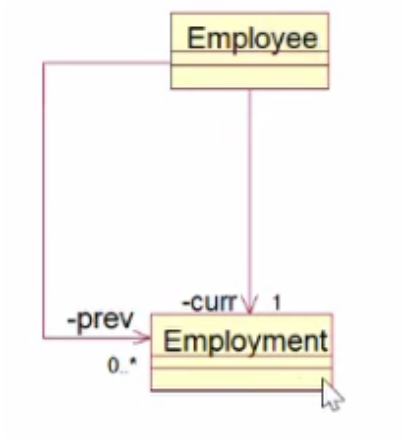
So this is navigation who owns the particular object
there are chances where one objects holds another objects data and another object holds this objects data
For Example we have a customer and the order, so customer will have the order information and order will in written have customer information
so it is **Bi Directional**
So to represent bi directional you should have to draw a line

3. **Role Name** In the UML whenever you see this diagram you can always name the object whatever you feel like you can call the object as a, b, c .....whatever
   But whenever you see a role name you have to create a object with the same name as what we have in the role name its cumpulsory

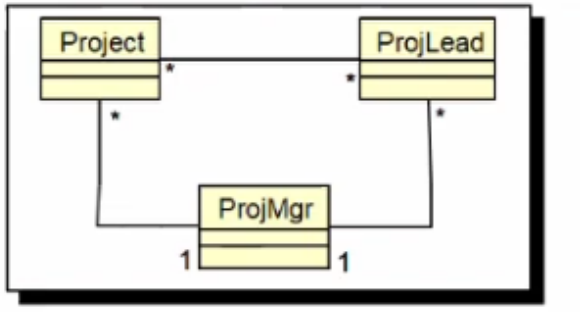
Has-a Relationship

- The 'Has-a' relationships are qualified by
  - Multiplicity
    - The number of instances with which a class is associated
    - Can be 1, 0..1, *, 1..*, 0..*, 2..*, 5..10, etc.
    - Multiplicity is by default 1
  - Navigability
    - Can be unidirectional or bidirectional
    - Navigability is by default bi-directional
  - Role name
    - The name of the instance in the relationship
    - Multiple 'has-a' based on different roles are possible





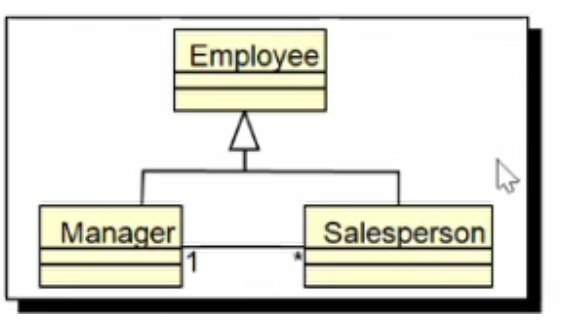
Now look at this i have a employee class and i have a employment
So, you can see 1 employee has a 1 current employment
and employee has previous 0 to many employment
How can a employee has 0 previous employment------if you are a fresher, obviously you have a 0 previous employment once you have one job, Once you enroll it then it will never be 0
Can anyone explain me this diagram?



This clearly says 1 project manager has many project lead
1 project manager has many projects to handle
and every project will have at least 1 manager
its bi directional project will have project manager information and project manager will have project information
here also it is bi directional project lead will have project manager, project manager will have project lead
So, when ever you are relating them you should go to particular object and from there you should find the relationship
Now, we have completed about project manager lets go to project
any project will have at least one project manager and many leads

when you come down to project lead, any project lead will have many projects to handle and will be reporting to one project manager

Try this



Here Manager is a employee

salesperson is a employee

1 manager will have many sales person

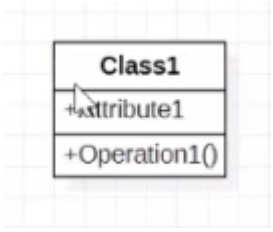
each sales person will report to one manager

You need to identify the classes, any classes will have three parts in it

When ever you draw a particular class, so class will have 3 parts in it

1. Class Name ---------and class name should always follow PascalCasing
2. Attributes

3. Members /Properties



So, whenever you are drawing you have to identify just class name don't bother about attributes and behaviour that is ok i don't mind

A trainer trains many trainees on a given technology in this course, which contains many modules – each module is comprised of different units and each unit has many topics.

- Identify the different classes from the above problem statement

Procedural approach

- Focus is on identifying **VERBS**
- Connections between functions established through Function Calls

OO approach

- Focus is on identifying **NOUNS**
- Connections between classes established through Relationships ('Is-a' and 'Has-a')

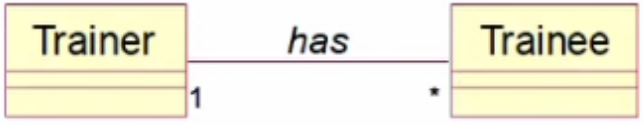
Identifying Classes

- Trainer
- Trainee
- Course
- Technology
- Module
- Unit
- Topic

- Identify the different connections (relationships) between the above classes

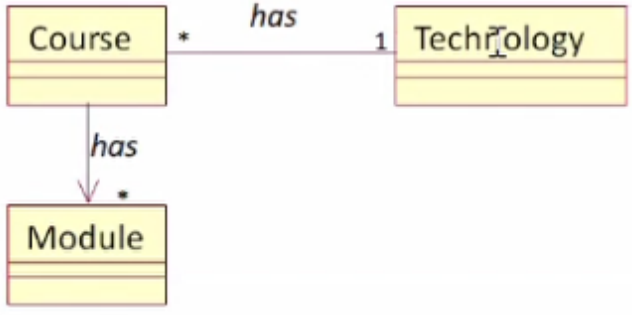
Identifying Relationships

- Trainer - Trainee
  - Trainer 'HAS' many Trainees
  - Every Trainee 'HAS' a Trainer



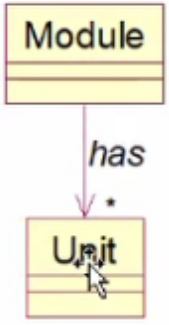
Identifying Relationships

- Course – Technology
- Course - Module
  - Course 'HAS' an associated Technology
  - A Technology has many courses
  - Course 'HAS' many Modules

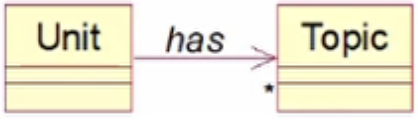


Identifying Relationships

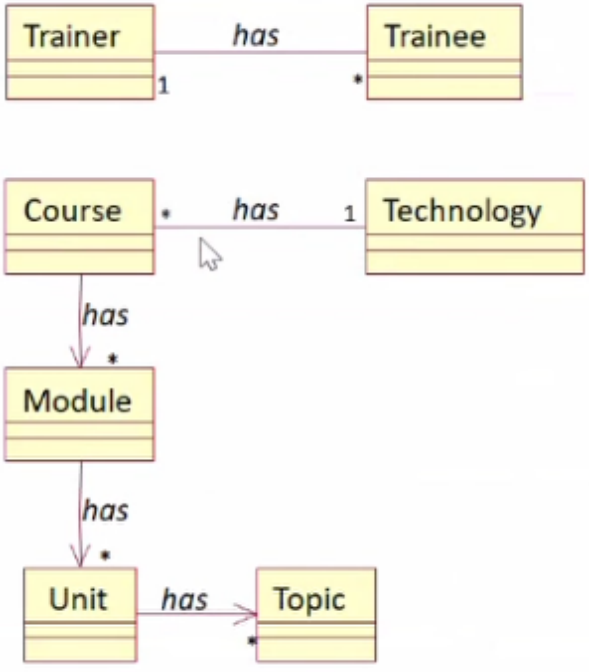
- Module – Unit
  - Module 'HAS' many Units



- Unit – Topic
  - Unit 'HAS' many Topics



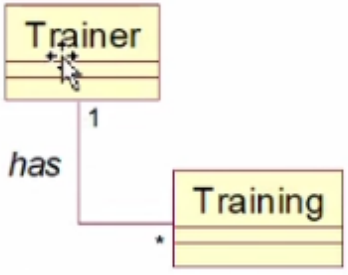
The OO Model



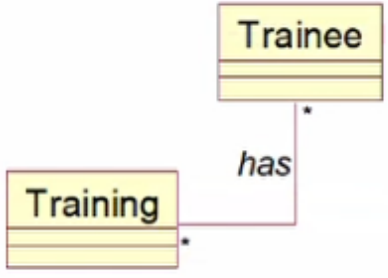
- How do you relate the Trainer & Trainee to the Course?

Conceptual Entity

- Trainer – Training
  - A Trainer (HAS) conducts many Trainings
  - A Training HAS a Trainer

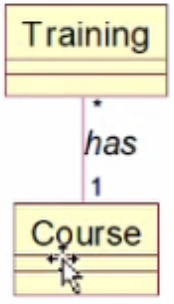


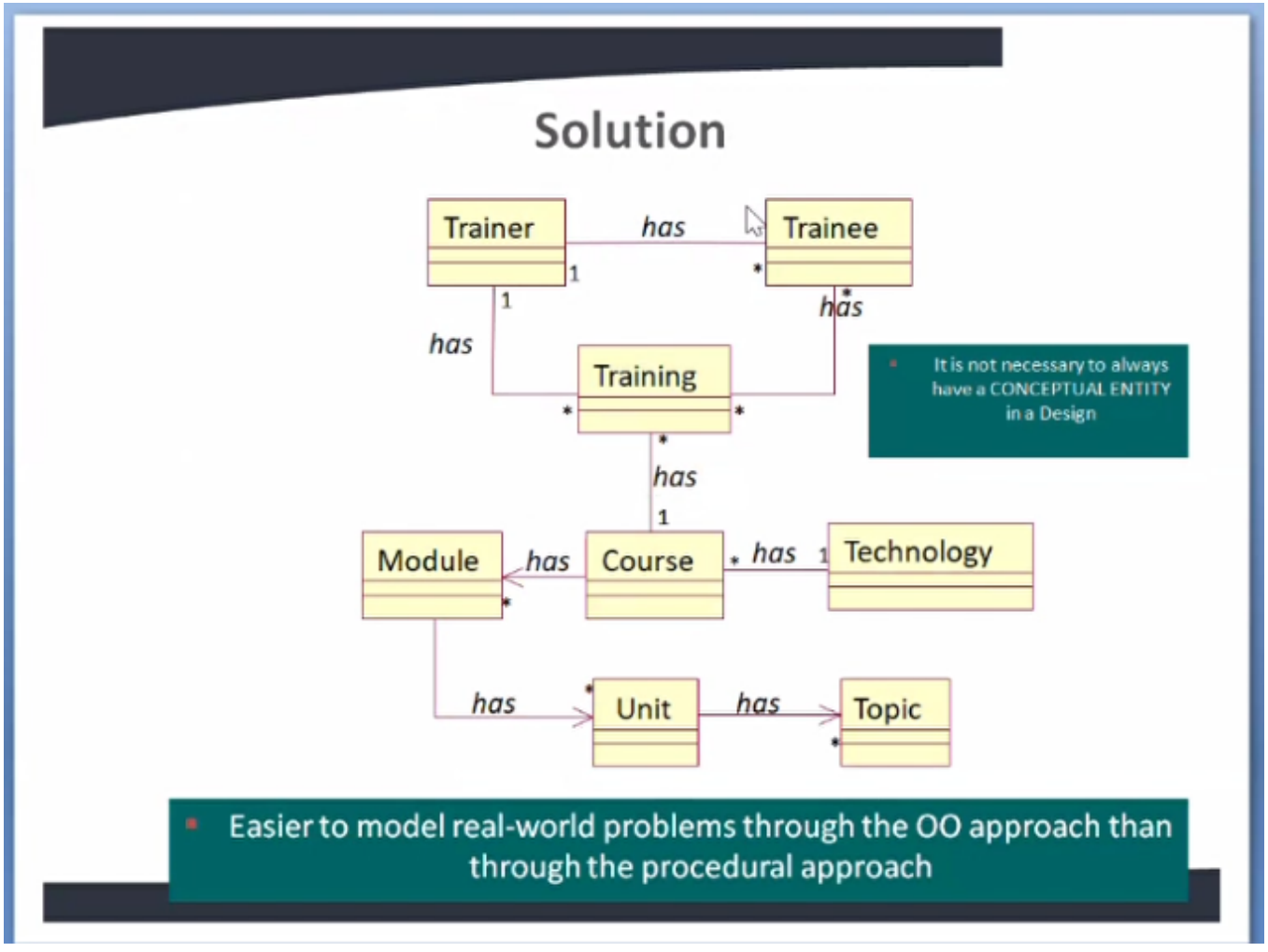
- Trainee – Training
  - A Trainee (HAS) attends many Trainings
  - A Training HAS a many Trainees



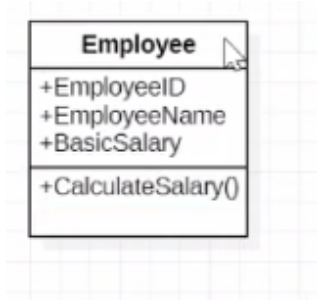
Conceptual Entity

- Training - Course
  - The Training (HAS) an association with a Course (conducted for a Course)
  - A Course HAS many Trainings





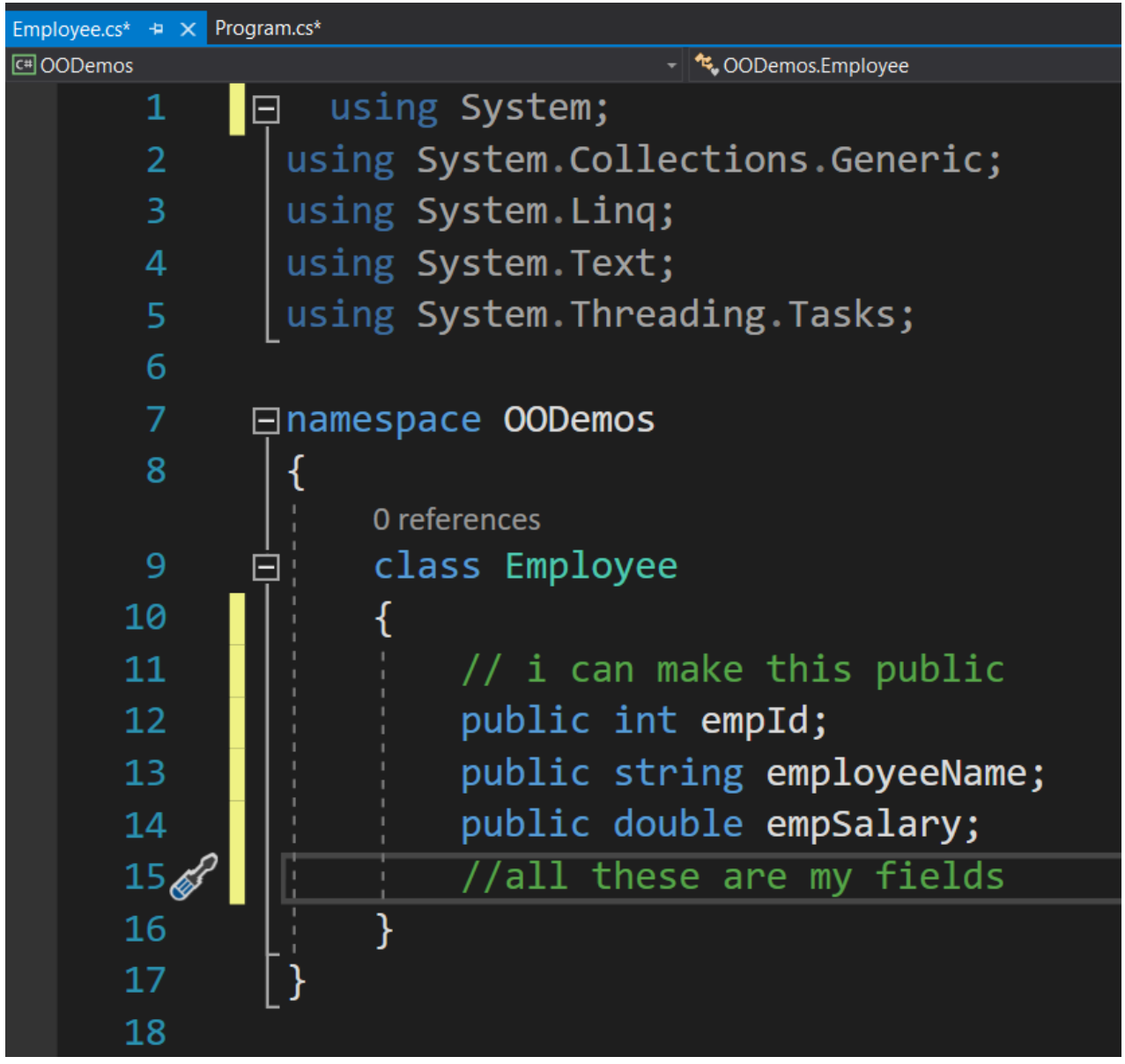
Now how do you write code for this



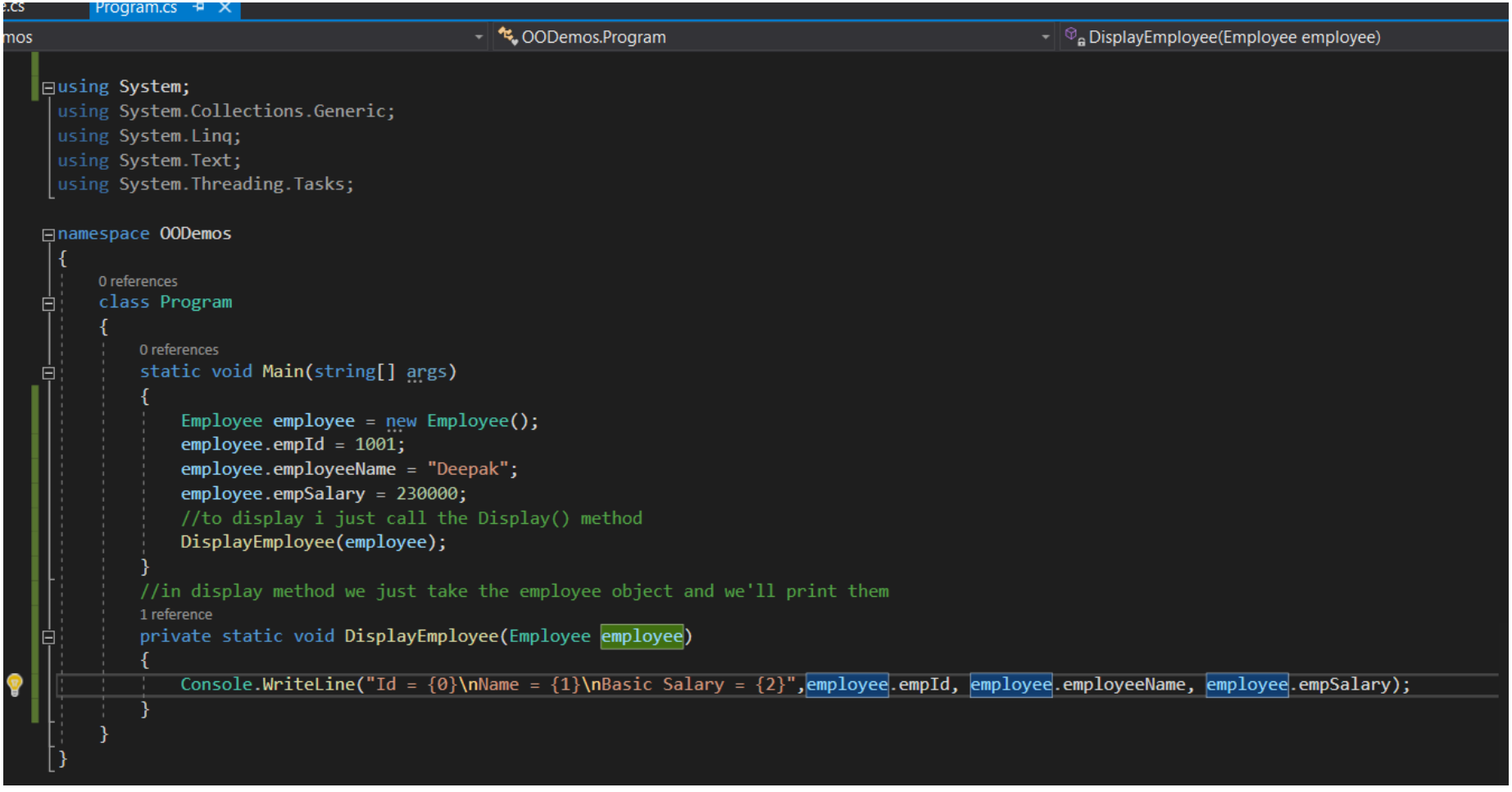
I have a UML design and i want to create a class for this

i Will create a new project named OODemo

I will create a Employee class now what is there in my employee class, Employee class should have property employeeid, employee name and basicsalary

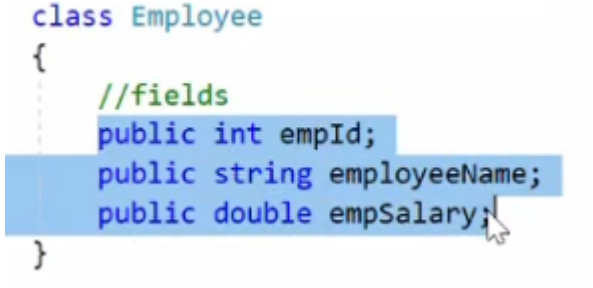

Employee.cs*
Program.cs*
OODemos
OODemos.Employee
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace OODemos
{
0 references
class Employee
{
// i can make this public
public int empId;
public string employeeName;
public double empSalary;
//all these are my fields
}
}

Now when i come down to my program, i want to create a instance of my employee

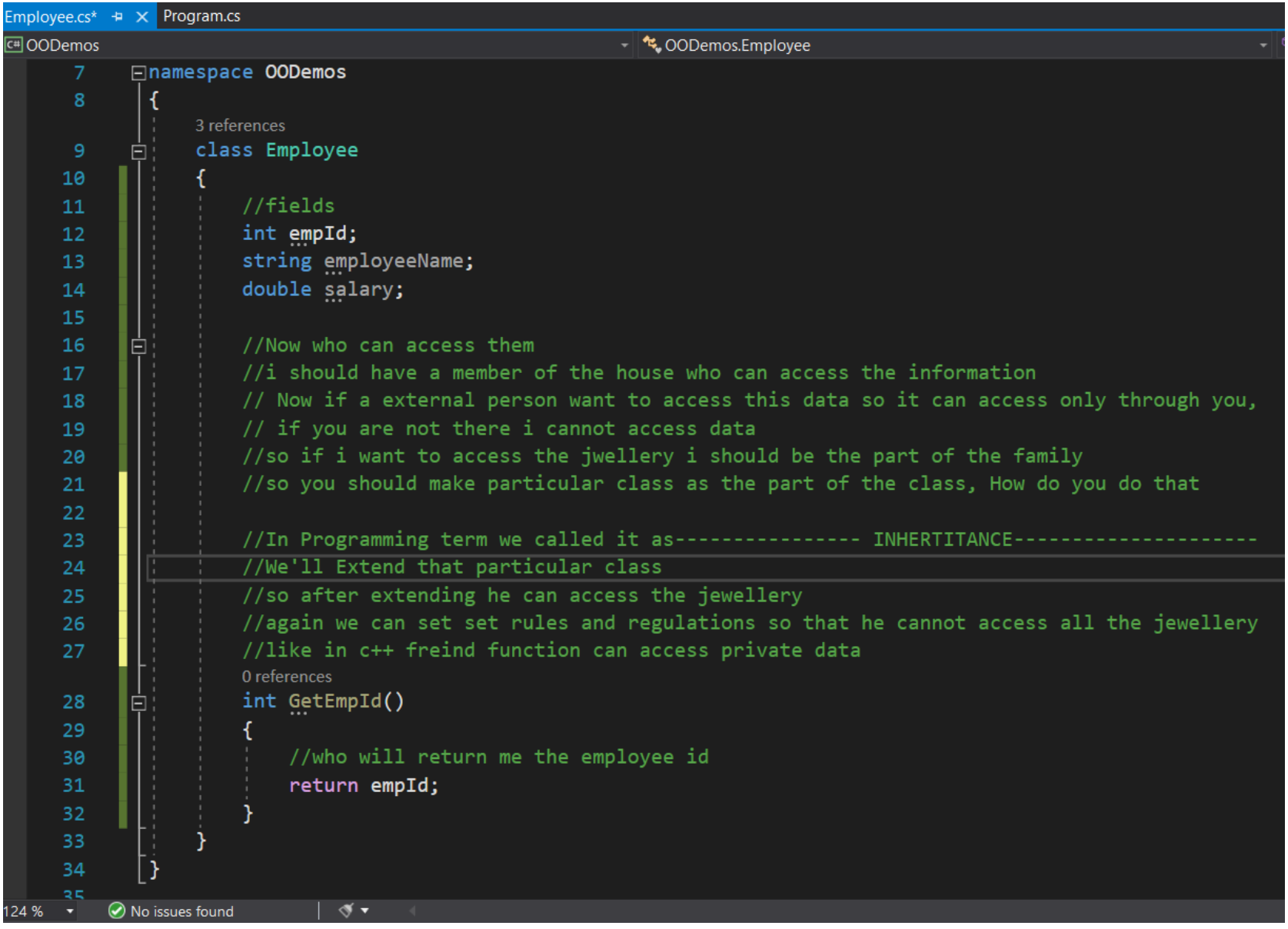

Program.cs
OODemos.Program
DisplayEmployee(Employee employee)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace OODemos
{
0 references
class Program
{
0 references
static void Main(string[] args)
{
Employee employee = new Employee();
employee.empId = 1001;
employee.employeeName = "Deepak";
employee.empSalary = 230000;
//to display i just call the Display() method
DisplayEmployee(employee);
}
//in display method we just take the employee object and we'll print them
1 reference
private static void DisplayEmployee(Employee employee)
{
Console.WriteLine("Id = {0}\nName = {1}\nBasic Salary = {2}",employee.empId, employee.employeeName, employee.empSalary);
}
}
}

Whats a difference between Property and field ?

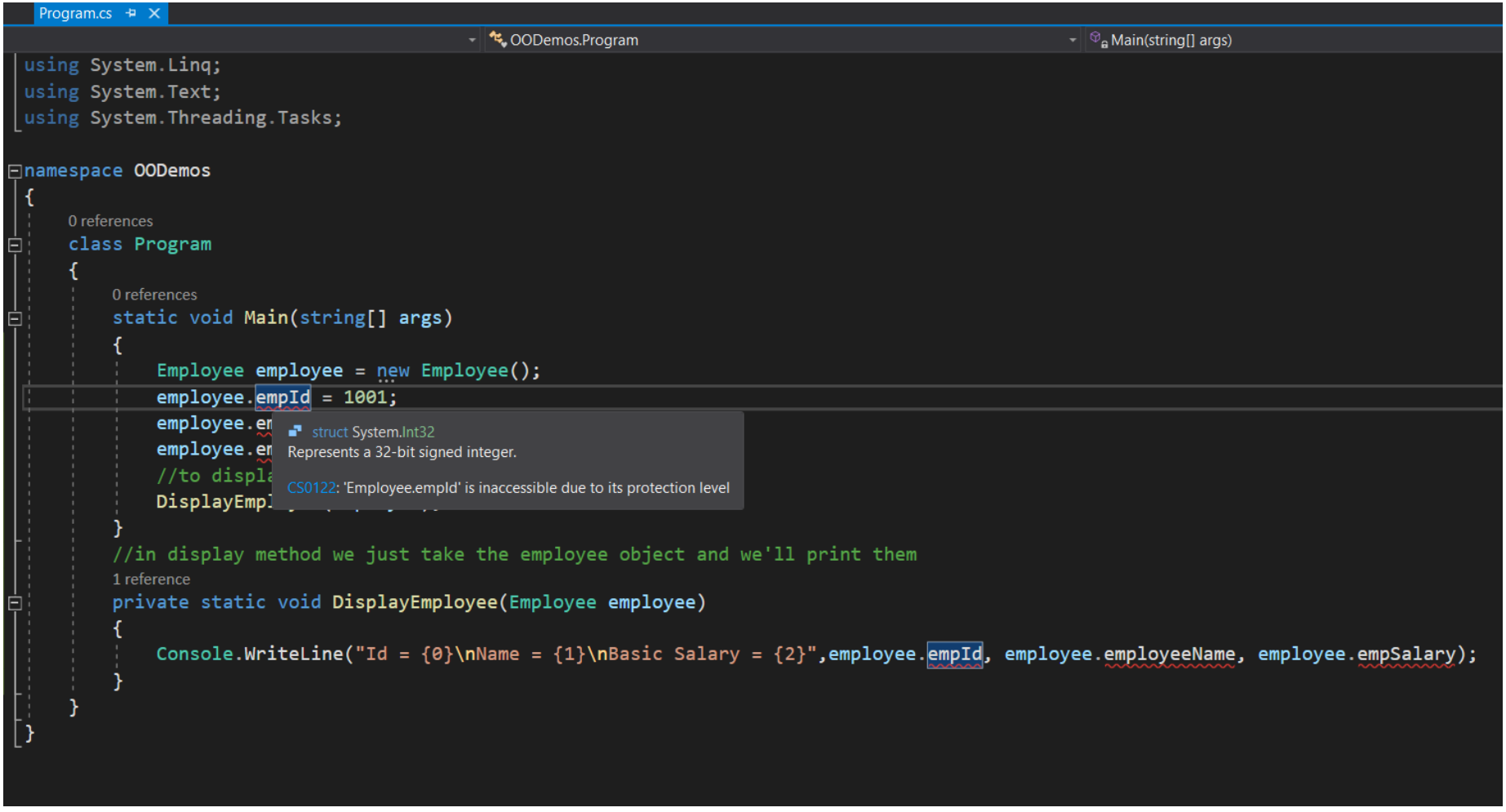
Now this is a bad design in Employee Class

1. 
```
class Employee
{
    //fields
    public int empId;
    public string employeeName;
    public double empSalary;
}
```

In Object oriented world we talk about security so these data should not be exposed directly they are like a jewellery, data can be accessed by member method only data should not be public, if you say public it will be like you are taking jewellery and putting it on the road


Employee.cs*
Program.cs
OODemos
OODemos.Employee
namespace OODemos
{
3 references
class Employee
{
//fields
int empId;
string employeeName;
double salary;
//Now who can access them
//i should have a member of the house who can access the information
// Now if a external person want to access this data so it can access only through you,
// if you are not there i cannot access data
//so if i want to access the jwellery i should be the part of the family
//so you should make particular class as the part of the class, How do you do that
//In Programming term we called it as---------- INHERTITANCE----------
//We'll Extend that particular class
//so after extending he can access the jewellery
//again we can set set rules and regulations so that he cannot access all the jewellery
//like in c++ freind function can access private data
0 references
int GetEmpId()
{
//who will return me the employee id
return empId;
}
}
}
124 %
No issues found

Now look at this you are getting errors saying boss you cannot access private data


Program.cs
OODemos.Program
Main(string[] args)
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace OODemos
{
0 references
class Program
{
0 references
static void Main(string[] args)
{
Employee employee = new Employee();
employee.empId = 1001;
employee.em
employee.em
//to displa
DisplayEmpl
}
//in display method we just take the employee object and we'll print them
1 reference
private static void DisplayEmployee(Employee employee)
{
Console.WriteLine("Id = {0}\nName = {1}\nBasic Salary = {2}",employee.empId, employee.employeeName, employee.empSalary);
}
}
}
struct System.Int32
Represents a 32-bit signed integer.
CS0122: 'Employee.empId' is inaccessible due to its protection level

Program.cs

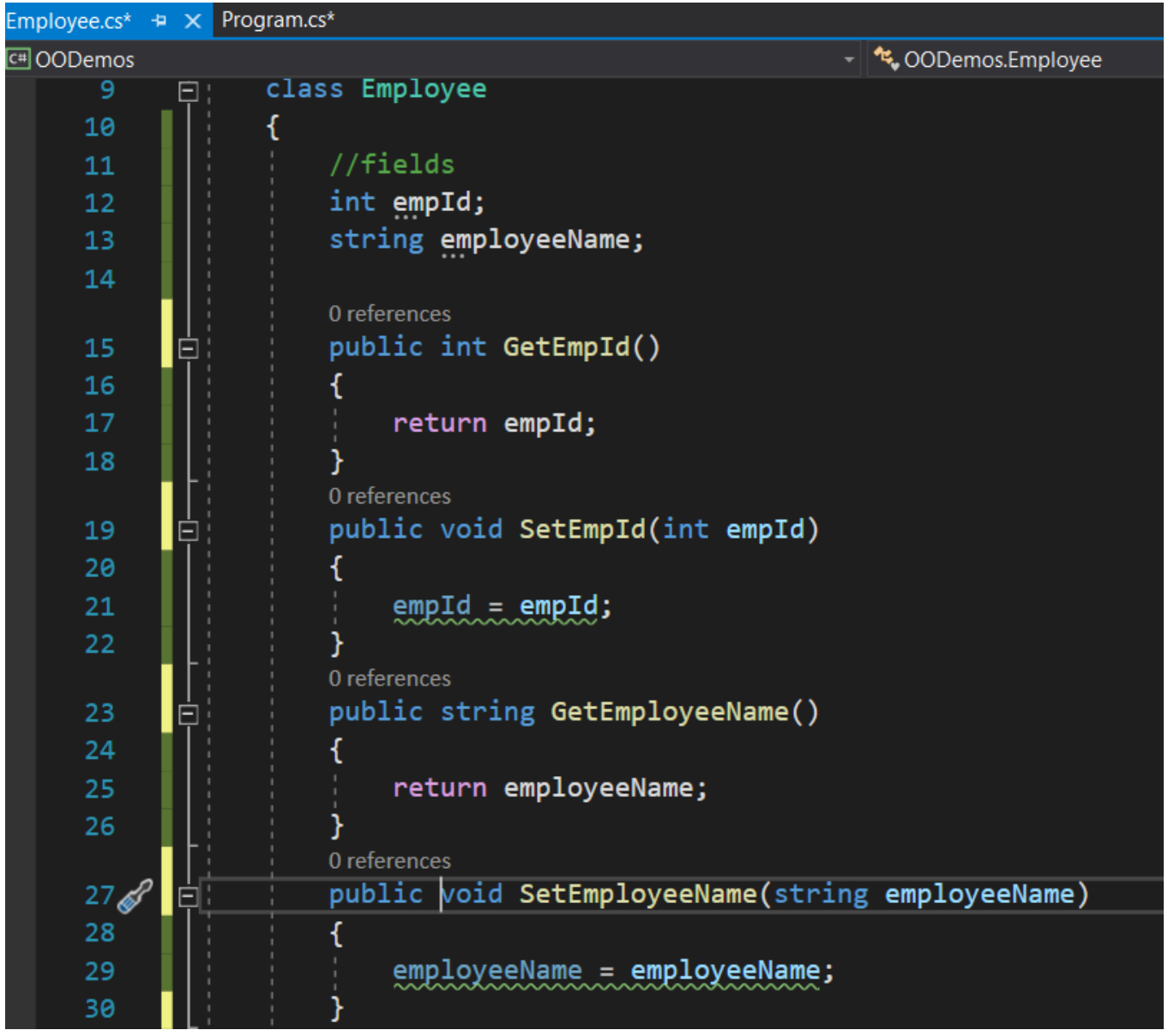
OODemos.Employee

```
namespace OODemos
{
    3 references
    class Employee
    {
        //fields
        int empId;
        string employeeName;
        double salary;

        //Now who can access them
        //i should have a member of the house who can access the information
        //Now when you are creating a members how will you access this particular id i want
        //to put some data to empId and get data from this

        //so you have a GetEmpId() to get empId
        0 references
        int GetEmpId()
        {
            //who will return me the employee id
            return empId;
        }
        //in the same way we have a SetEmpId() which will take a int empId and we'll set it
        0 references
        void SetEmpId(int empId)
        {
            empId = empId;
        }

        //the same thing we have to do it for name
    }
```

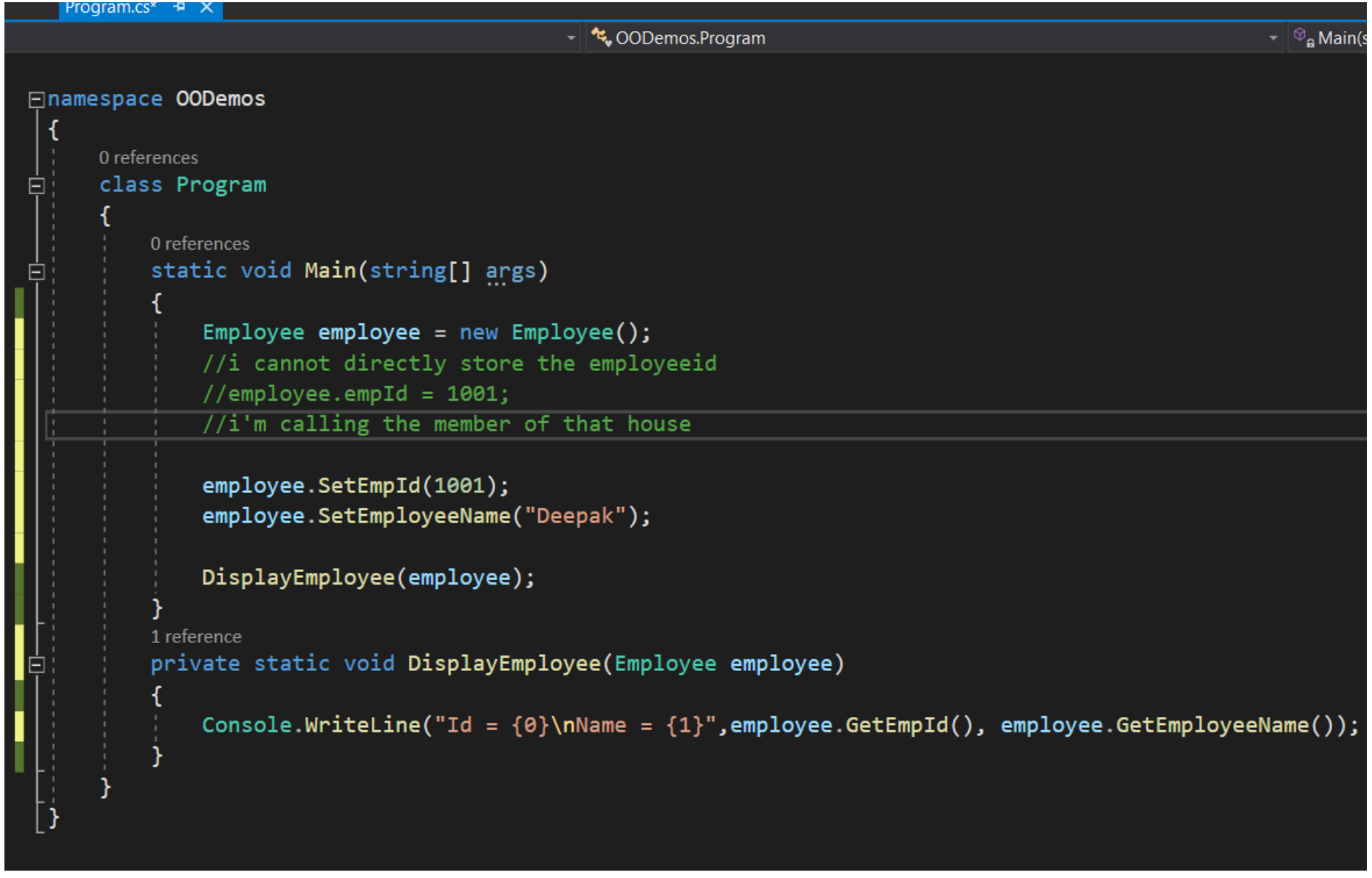
Program.cs

OODemos.Employee

```
class Employee
{
    //fields
    int empId;
    string employeeName;

    0 references
    int GetEmpId()
    {
        return empId;
    }
    0 references
    void SetEmpId(int empId)
    {
        empId = empId;
    }
    0 references
    string GetEmployeeName()
    {
        return employeeName;
    }
    0 references
```


Employee.cs*
Program.cs*
OODemos
OODemos.Employee
class Employee
{
//fields
int empId;
string employeeName;
0 references
public int GetEmpId()
{
return empId;
}
0 references
public void SetEmpId(int empId)
{
empId = empId;
}
0 references
public string GetEmployeeName()
{
return employeeName;
}
0 references
public void SetEmployeeName(string employeeName)
{
employeeName = employeeName;
}

We make member methods as public so that we can access them

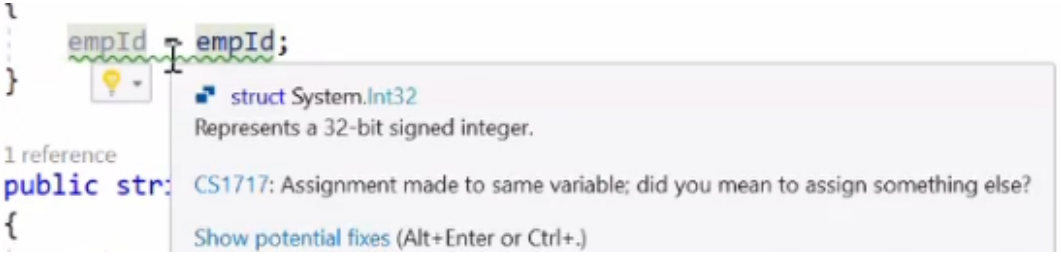

Program.cs*
OODemos.Program
Main(s
namespace OODemos
{
0 references
class Program
{
0 references
static void Main(string[] args)
{
Employee employee = new Employee();
//i cannot directly store the employeeid
//employee.empId = 1001;
//i'm calling the member of that house
employee.SetEmpId(1001);
employee.SetEmployeeName("Deepak");
DisplayEmployee(employee);
}
1 reference
private static void DisplayEmployee(Employee employee)
{
Console.WriteLine("Id = {0}\nName = {1}",employee.GetEmpId(), employee.GetEmployeeName());
}
}
}

```
Id = 0
Name = Deepak
Press any key to continue . . .
```

```
1 reference
public void SetEmpId(int empId)
{
    empId = empId;
}
```

when we double click on empId look at this all 3 gets heighlighted

and there is a green underline, never ignore the green line



when you move your mouse pointer what it says , means you are taking this empId and you are assigning back to it, but that is not the idea right

```
3 references
class Employee
{
    //fields
    int empId;
    string employeeName;
    //double empSalary;

    1 reference
    public int GetEmpId()
    {
        return empId;
    }
    1 reference
    public void SetEmpId(int empId)
    {
        empId = empId;
    }

    1 reference
    public string GetEmpName()
    {
        return employeeName;
    }
```

2. then i want to assign this to objects empid

1. i want to take the empid from here and

3. so how do you say that this particular empid belongs to the object

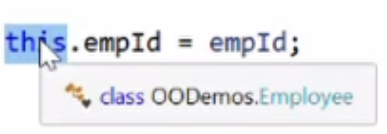
By saying **this.empid**

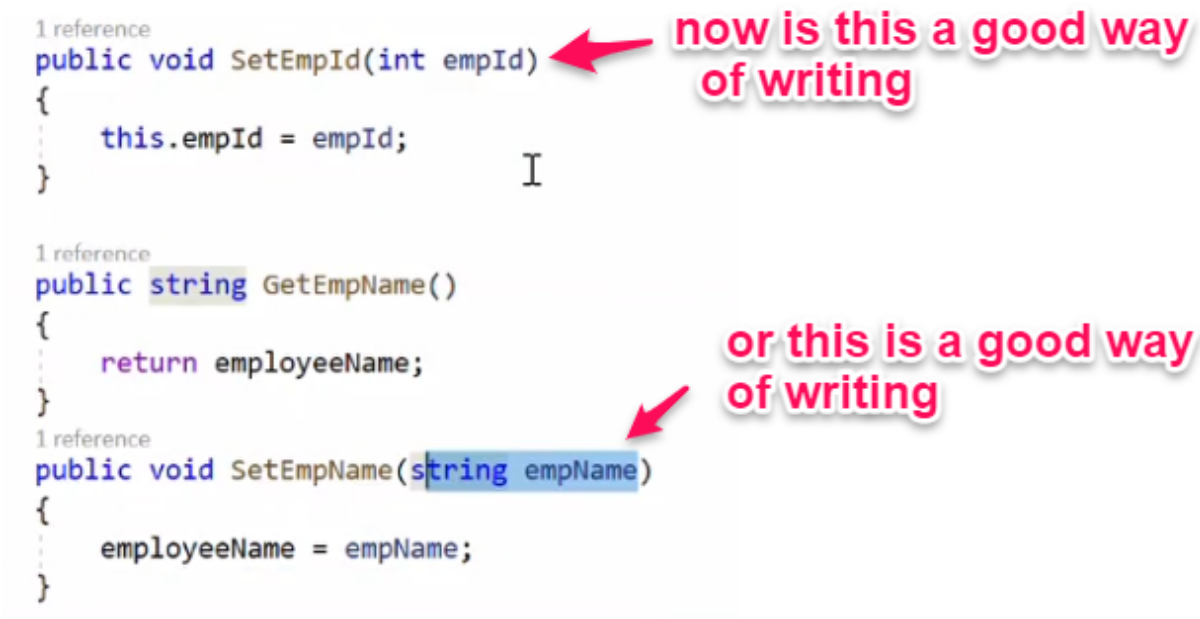
```
1 reference
public void SetEmpId(int empId)
{
    this.empId = empId;
}
```

When you say this, so this represent the current object, the object who ever is calling this one

```
Employee employee = new Employee();
employee.SetEmpId(1001);
employee.SetEmpName("Shashi");
//employee.empSalary = 230000;
```

in our case it is the employee object

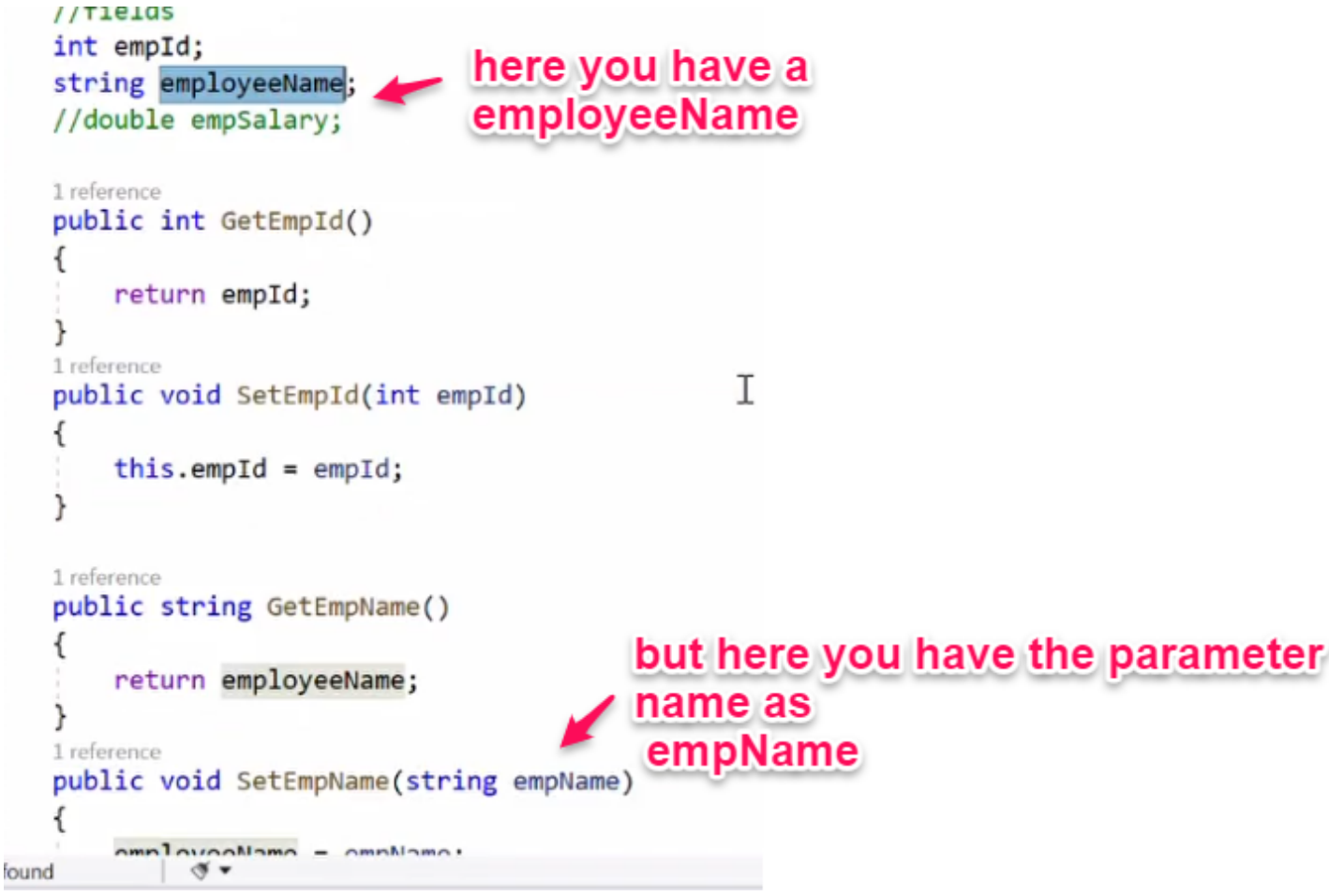




what is the difference

in 1 st one my instance id is empid there we have the same empid

but in 2nd case

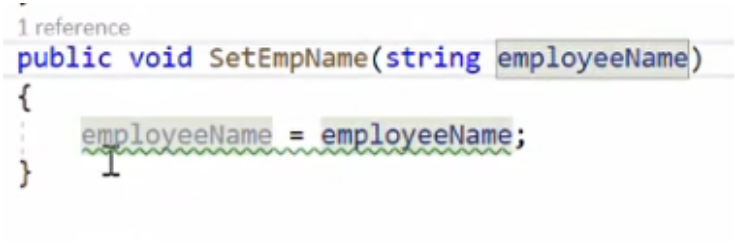


So should it be employeeName or empName

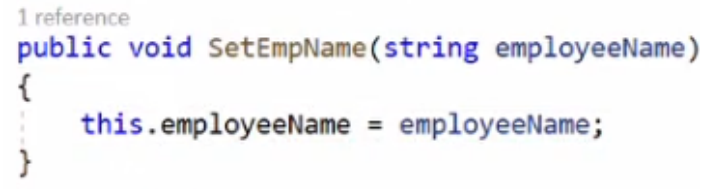
----------employeeName

Why employeeName ?

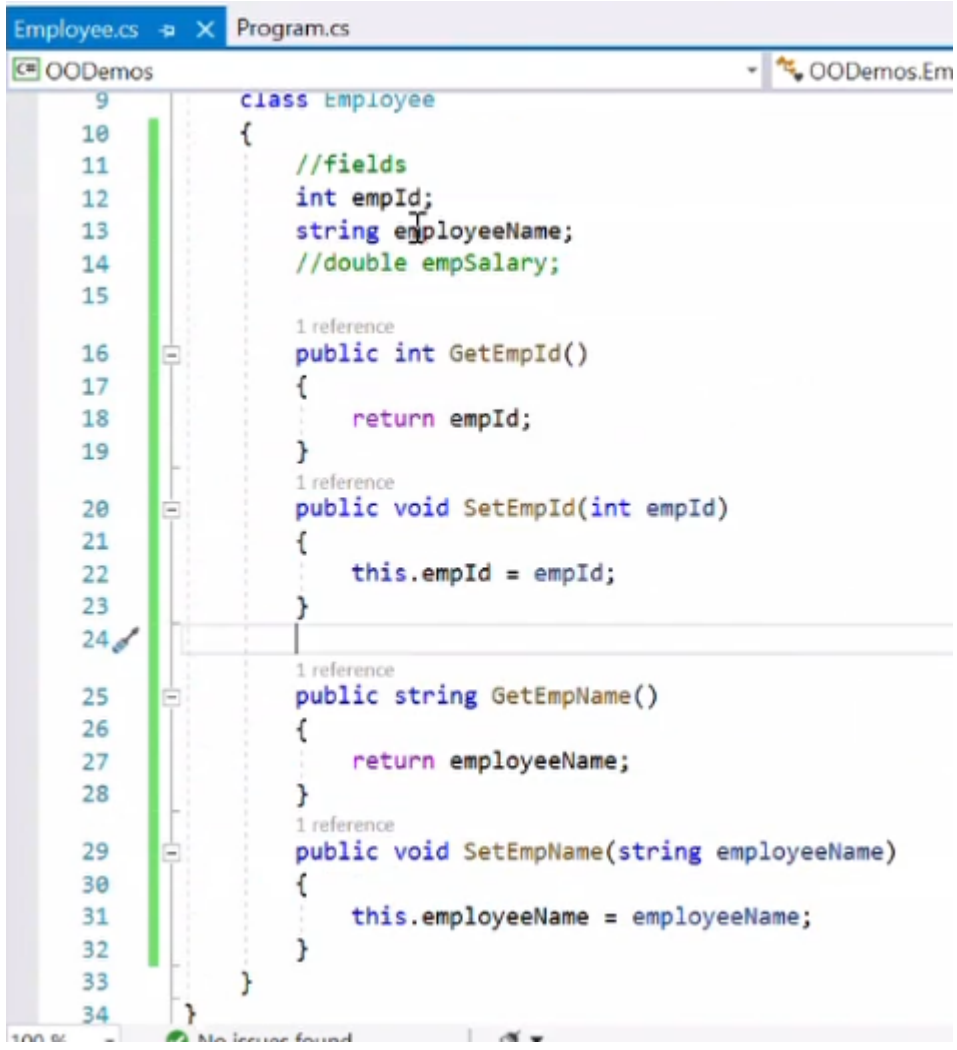
why not abc?
because whenever somebody reads this they should also understand what exactly it means to where we are trying to store this data thats a reason this particular variable name should be same



Now as this is same you can see when i double click on any on e all this becomes heighlight (local)
so you should always say boss this belongs to the object, so you should always says this.

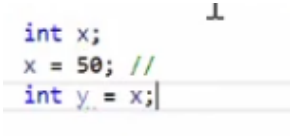


2nd What is the problem in this particular program ?



Now imagine this is a simple program and you have 2 fields and for this 2 fields you are writing soo many set and get methods

for every field you have 1 set and 1 get method what if we have 100 or 200 fields we have to write soo many methods it dosent make any sence
So java people still struggling with the same thing, in java world we still do the same thing
but thanks to ide if you are using ide you can generate these set get methods easily, microsoft thought about this
like
How do you store the value to a variable



So based on weather the variable is on left hand side or the right hand side of your equals operator it will either have assigning or fetching the data
now this is easy this is what we are doing from soo many decades
now look at this

```
Employee employee = new Employee();
employee.SetEmpId(1001);
employee.SetEmpName("Shashi");
//employee.empSalary = 230000;

DisplayEmployee(employee);
```

this set get dosent make any sence this is soo painfull for the developer

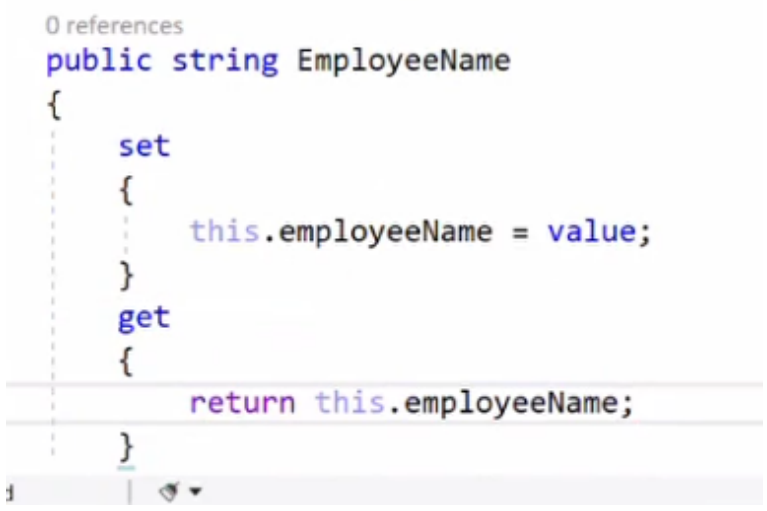
So Microsoft introduced something called Property

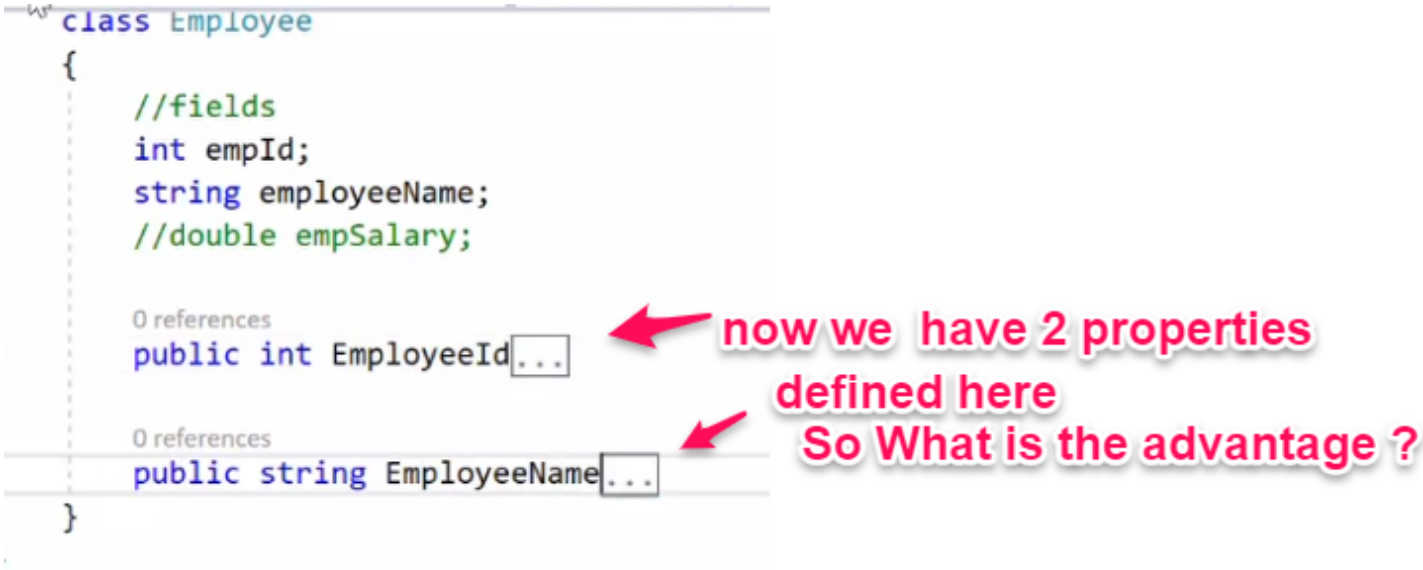
What is property?

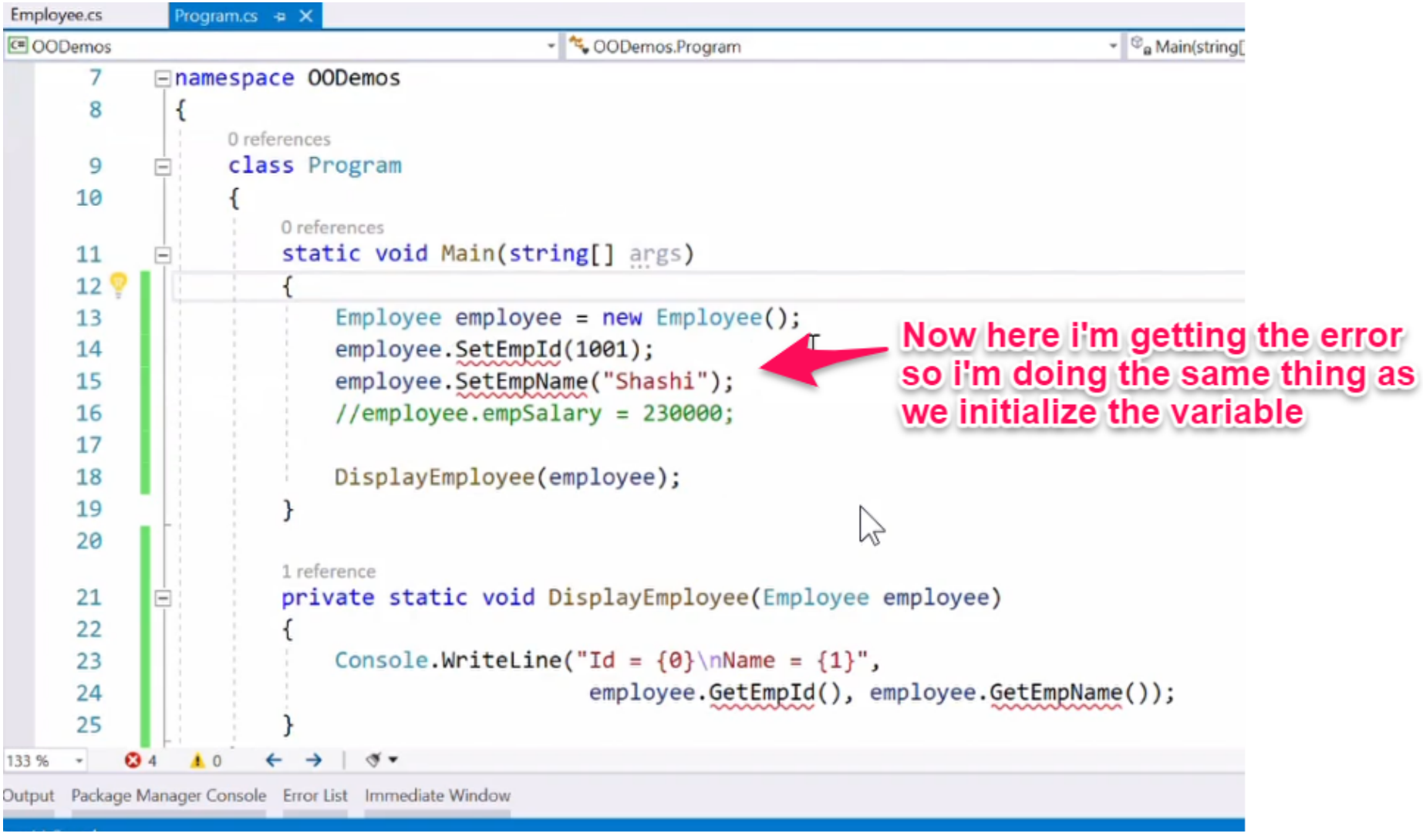
the same get set method what you are writing here

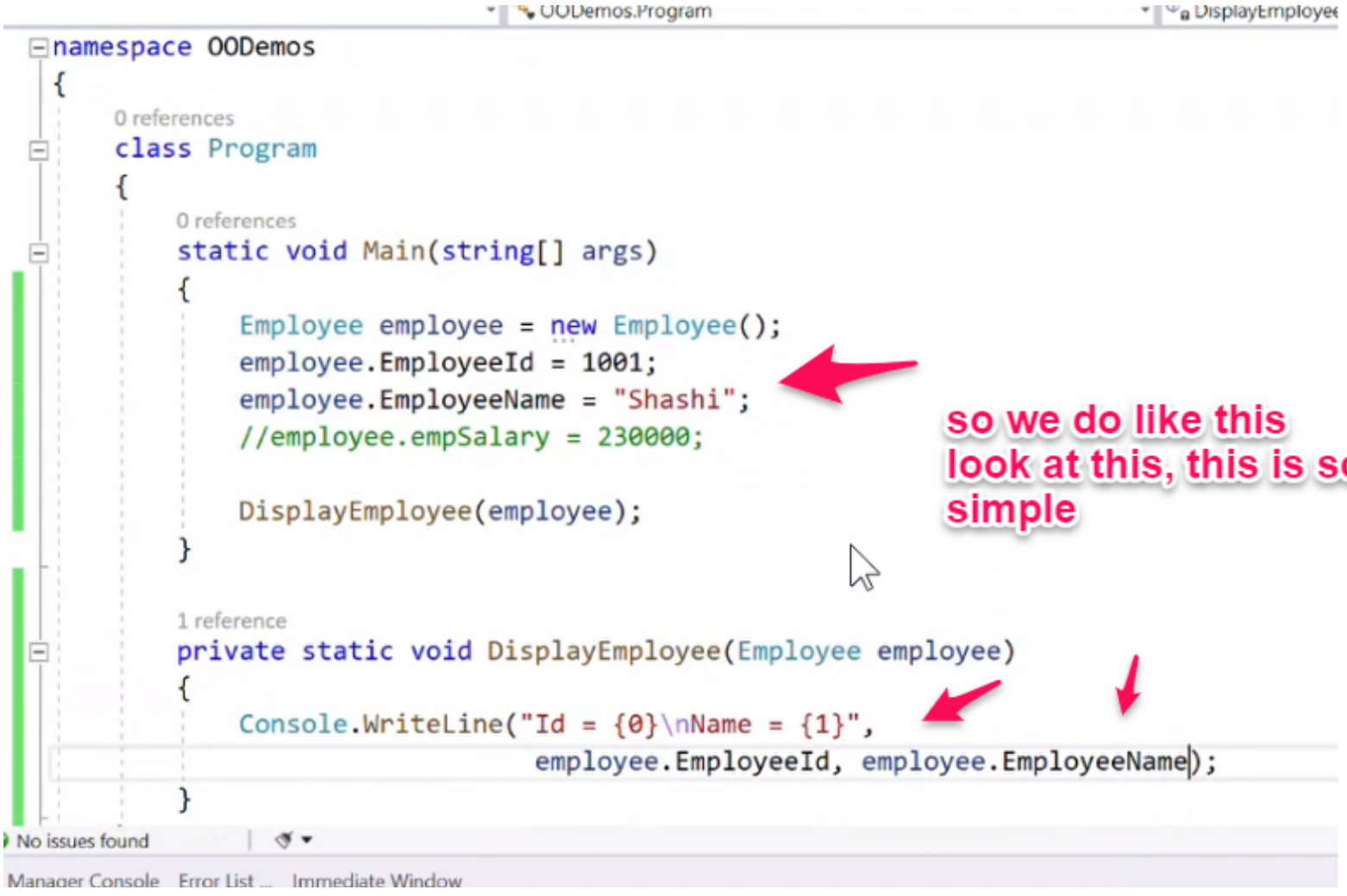
```
public int EmployeeId
{
    set
    {
        this.empId = value;
    }
    get
    {
        return this.empId;
    }
}
```

We write like this same thing we do for employee name



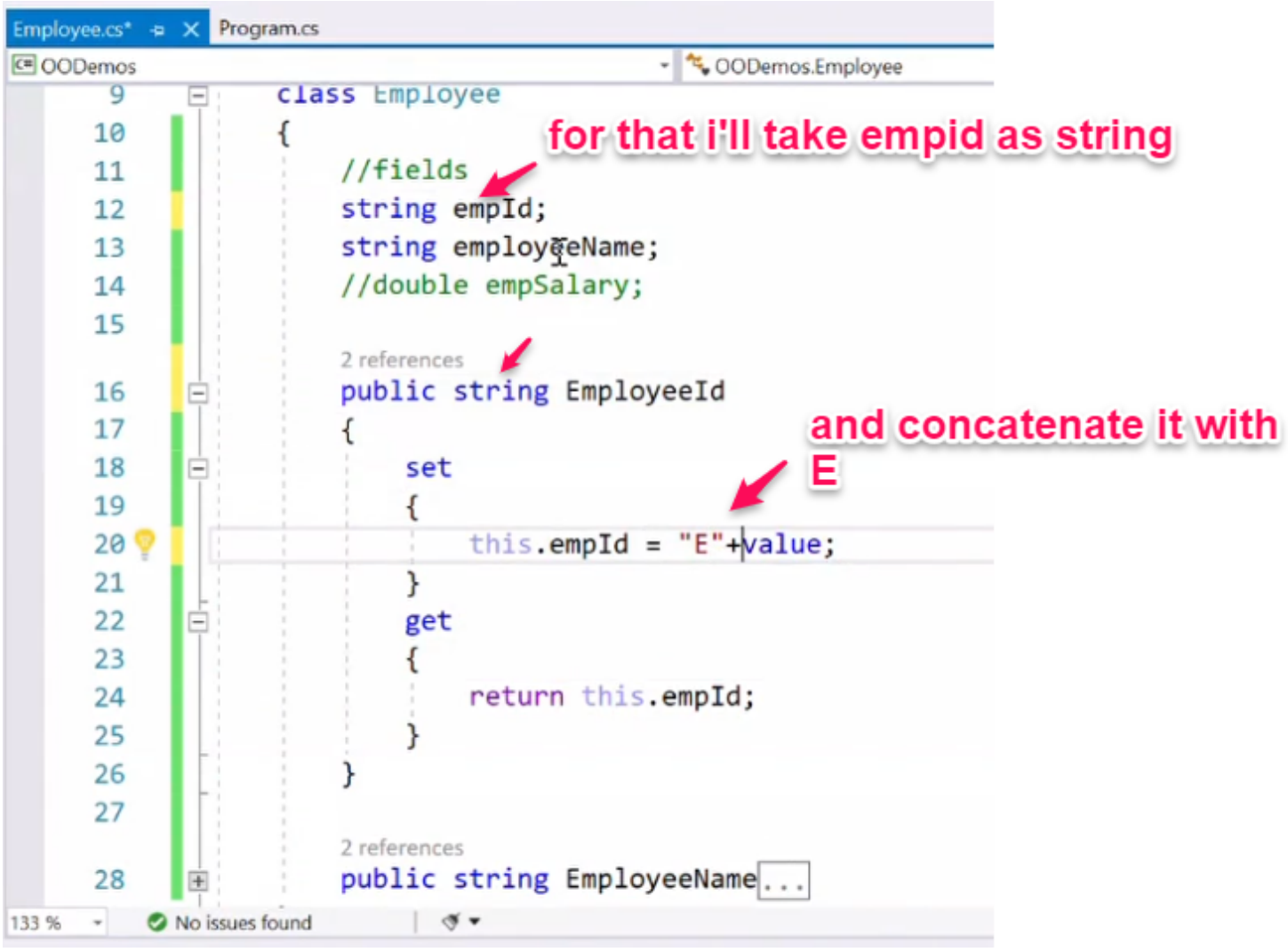

class Employee
{
//fields
int empId;
string employeeName;
//double empSalary;
0 references
public int EmployeeId...
0 references
public string EmployeeName...
}
now we have 2 properties
defined here
So What is the advantage ?

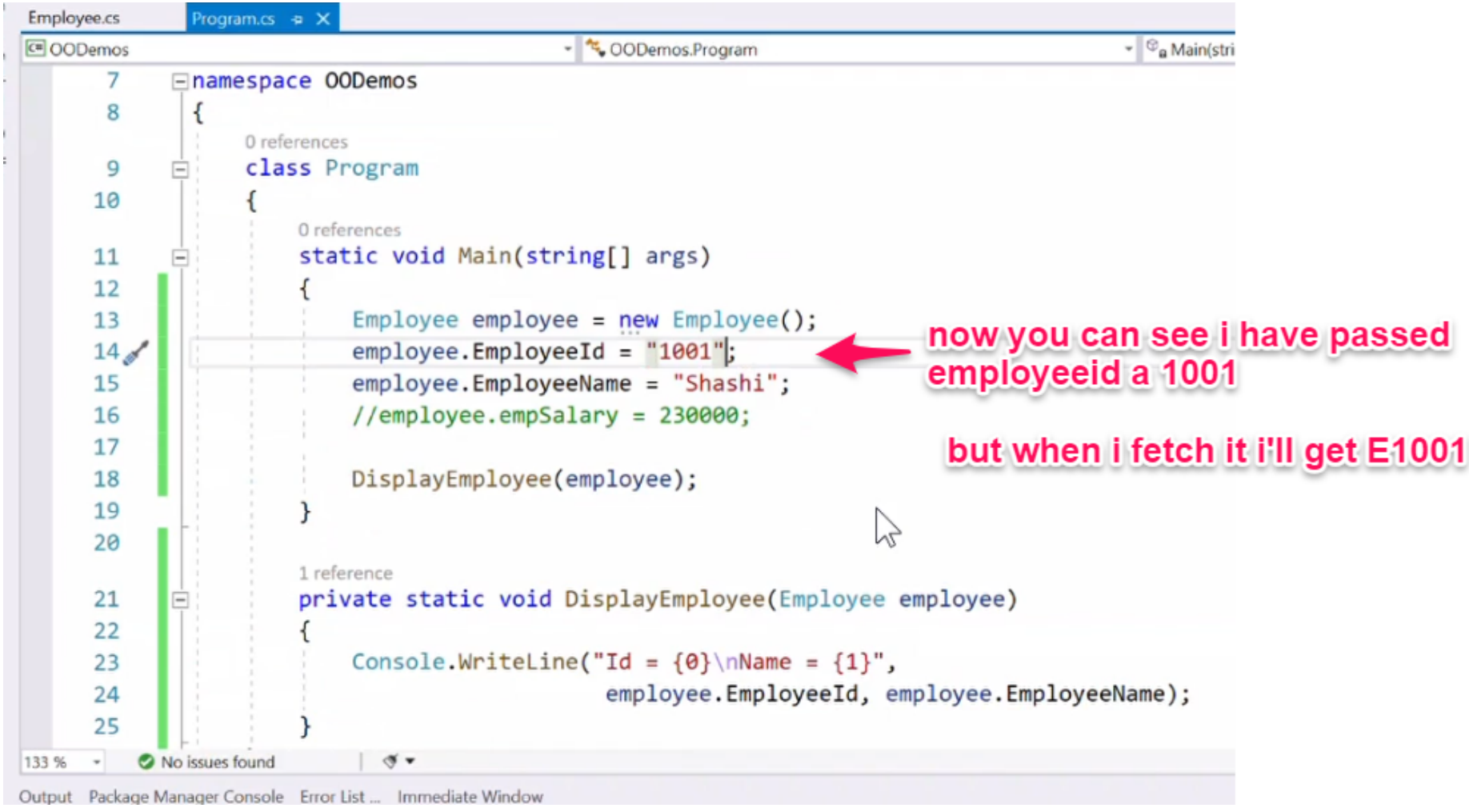

Employee.cs
Program.cs
OODemos
OODemos.Program
Main(string[
namespace OODemos
{
0 references
class Program
{
0 references
static void Main(string[] args)
{
Employee employee = new Employee();
employee.SetEmpId(1001);
employee.SetEmpName("Shashi");
//employee.empSalary = 230000;
DisplayEmployee(employee);
}
1 reference
private static void DisplayEmployee(Employee employee)
{
Console.WriteLine("Id = {0}\nName = {1}",
employee.GetEmpId(), employee.GetEmpName());
}
Now here i'm getting the error
so i'm doing the same thing as
we initialize the variable
133 %
4
0
Output
Package Manager Console
Error List
Immediate Window

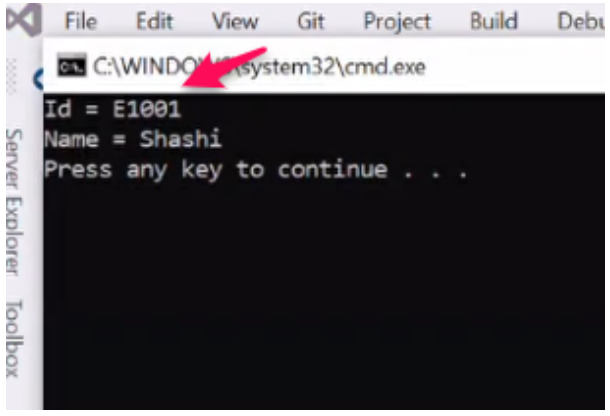

OODemos.Program
DisplayEmployee
namespace OODemos
{
0 references
class Program
{
0 references
static void Main(string[] args)
{
Employee employee = new Employee();
employee.EmployeeId = 1001;
employee.EmployeeName = "Shashi";
//employee.empSalary = 230000;
DisplayEmployee(employee);
}
1 reference
private static void DisplayEmployee(Employee employee)
{
Console.WriteLine("Id = {0}\nName = {1}",
employee.EmployeeId, employee.EmployeeName);
}
so we do like this
look at this, this is soo
simple
No issues found
Manager Console
Error List ...
Immediate Window

So PROPERTY are those which will have the set get methods for a field

Now what if i want that employee id always starts with E while storing in database


Employee.cs*
Program.cs
OODemos
OODemos.Employee
for that i'll take empid as string
and concatenate it with
E
class Employee
{
//fields
string empId;
string employeeName;
//double empSalary;
2 references
public string EmployeeId
{
set
{
this.empId = "E"+value;
}
get
{
return this.empId;
}
}
2 references
public string EmployeeName...
133 %
No issues found


Employee.cs
Program.cs
OODemos
OODemos.Program
Main(stri
namespace OODemos
{
0 references
class Program
{
0 references
static void Main(string[] args)
{
Employee employee = new Employee();
employee.EmployeeId = "1001";
employee.EmployeeName = "Shashi";
//employee.empSalary = 230000;
DisplayEmployee(employee);
}
1 reference
private static void DisplayEmployee(Employee employee)
{
Console.WriteLine("Id = {0}\nName = {1}",
employee.EmployeeId, employee.EmployeeName);
}
now you can see i have passed employeeid a 1001
but when i fetch it i'll get E1001
133 %
No issues found
Output
Package Manager Console
Error List ...
Immediate Window

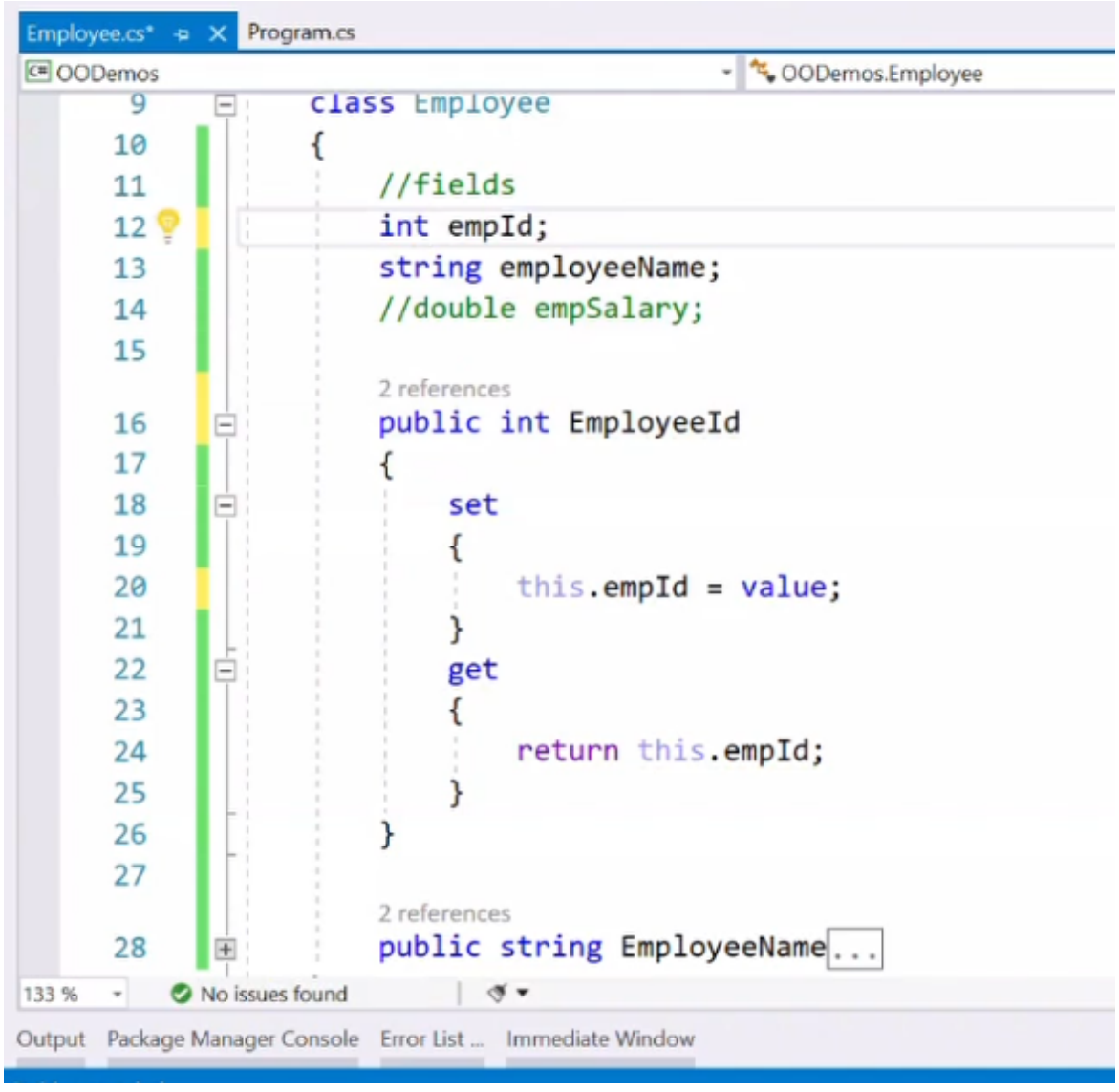


So i can manupulate on this particular data

i take a particular data i can check weather it is less than or greater than i can do lot of things here

so this is the advantage of creating that as a property

But if you dont want to manupulate the data if it is a straight forward code

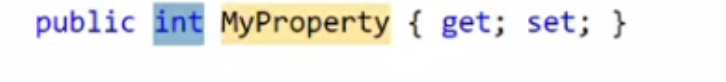

Employee.cs*
Program.cs
OODemos
OODemos.Employee
class Employee
{
//fields
int empId;
string employeeName;
//double empSalary;
2 references
public int EmployeeId
{
set
{
this.empId = value;
}
get
{
return this.empId;
}
}
2 references
public string EmployeeName...
133 %
No issues found
Output
Package Manager Console
Error List ...
Immediate Window

like this i'm storing the int value and i'm fetching the int value
so if it is straight forward then you no need of writing soo many lines of code

SO microsoft said boss you dont write all this rubbish code allow me to writing all this rubbish code

So to create a property you just have to write

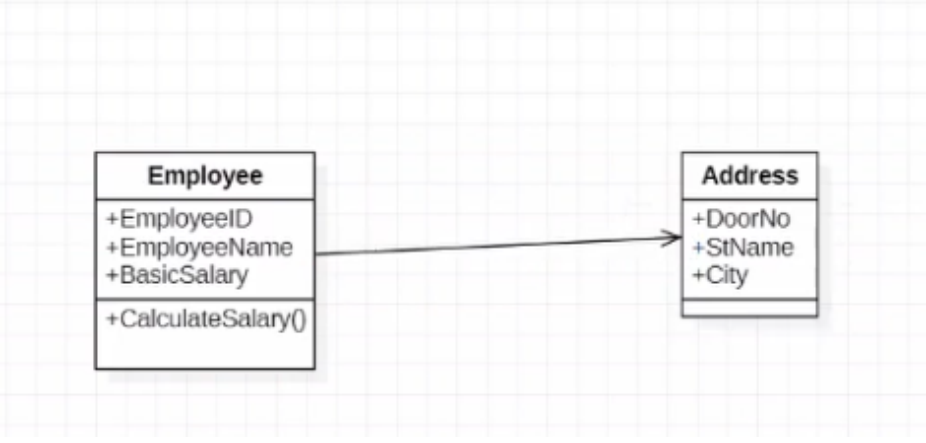
prop and press tab tab



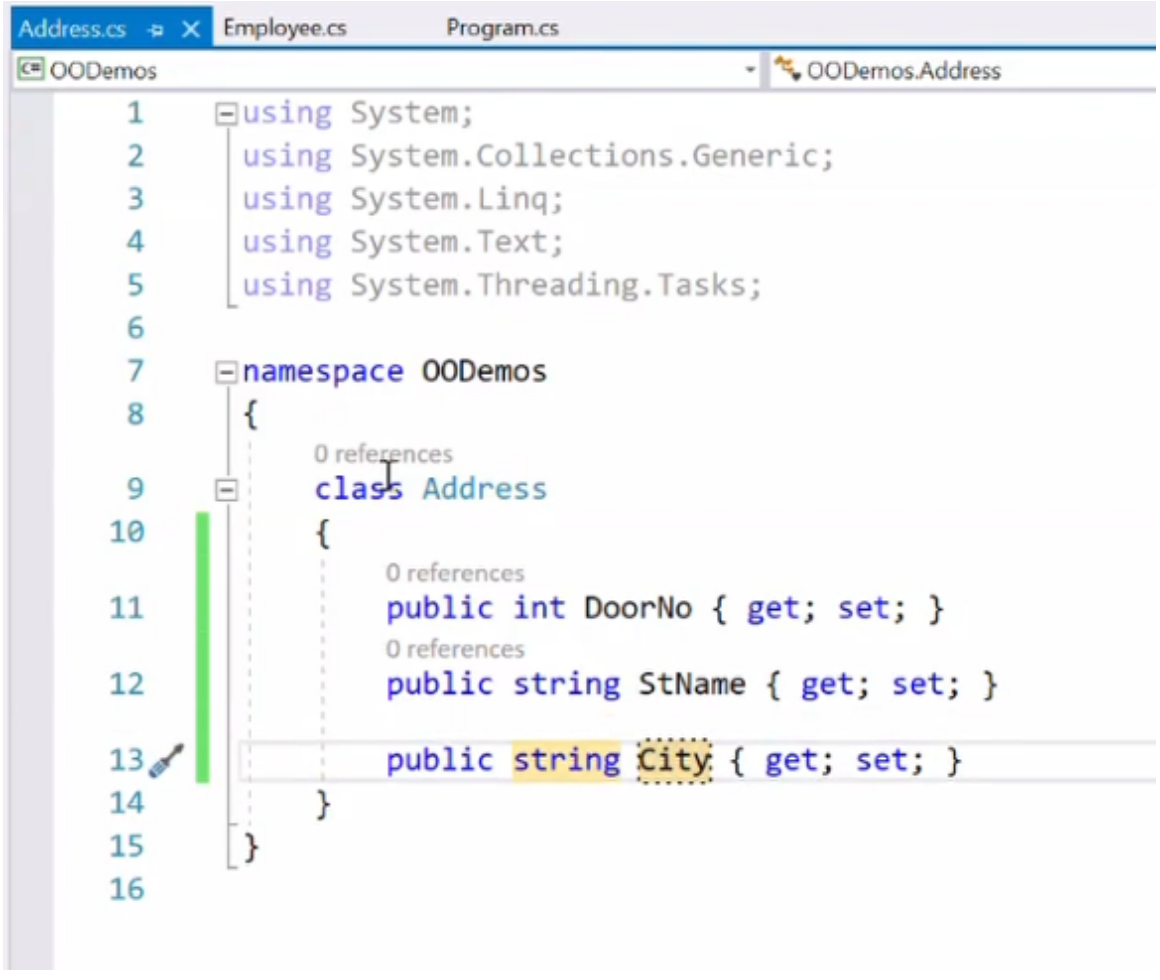
and edit accordingly

```
2 references
public int EmployeeId { get; set; }
2 references
public string EmployeeName { get; set; }
```

that's it, look how easy this is,
now internally we have a private integer variable will be created and public method will be created which will have set get method so all that 4 line of code is encapsulated in these 1 line of code, i'll not sit and write the whole code but my compiler do it for me this is the power of .net
now my employee has a address

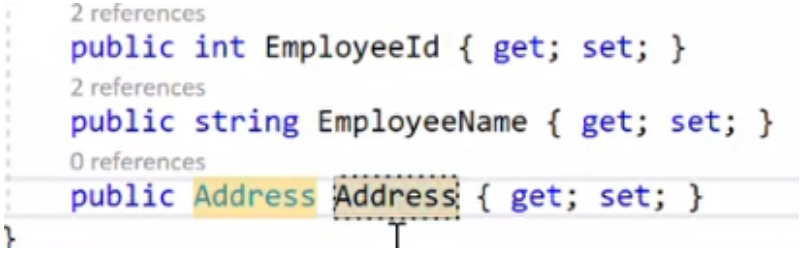


So Is it uni direction or bi direction ?---------->uni, employee owns the address

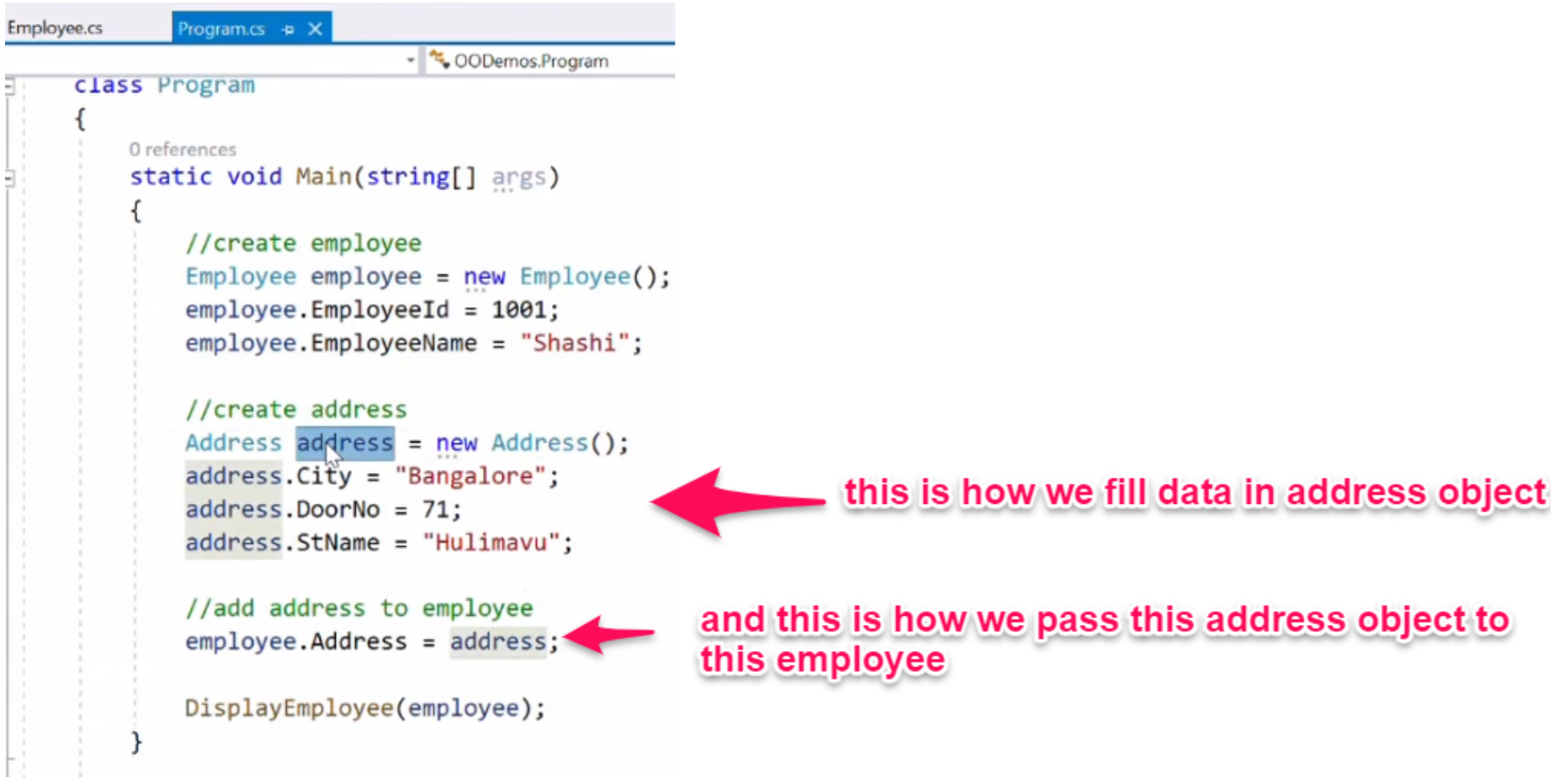


So we have created a model class named address with 3 properties

Now how will you say employee has address? ---------just like any other variable what we have here



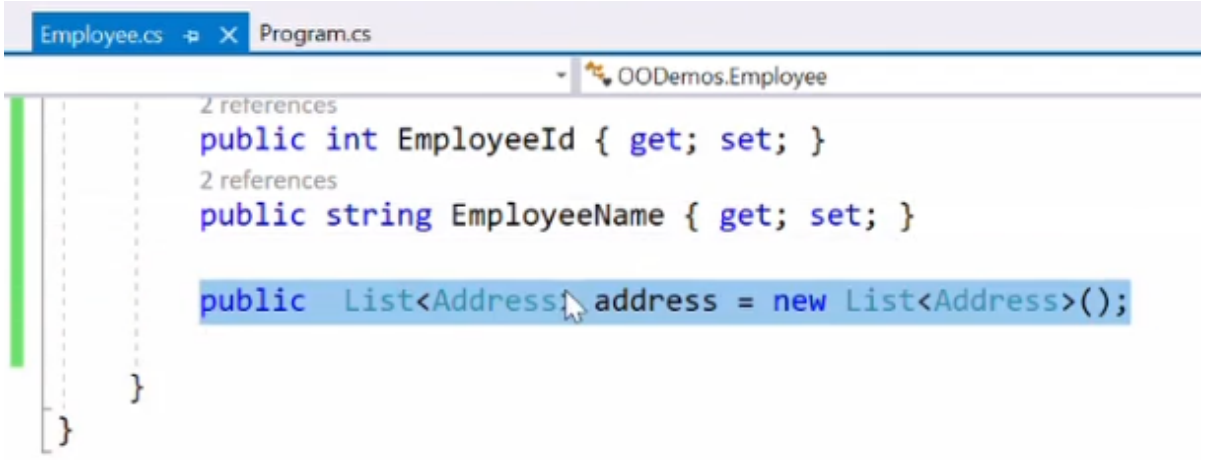
so just like empid empname we have empaddress



Now this is for one address but see in amazon we can gave multiple address, so an employee can have multiple address

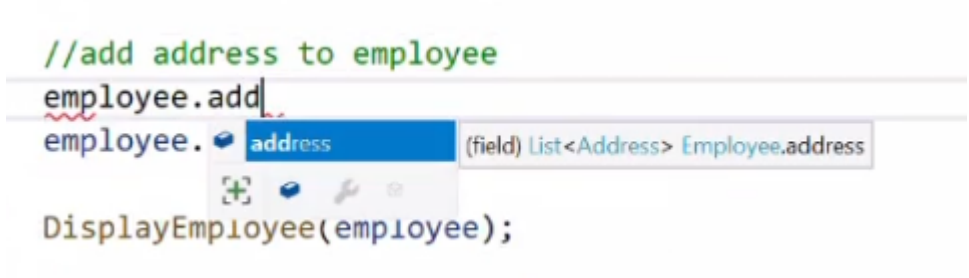
So how do you handle if there is a multiple address

So to achieve in employee we have to create list of object



now see in program you cannot add address directly

look at this when i say employee.address



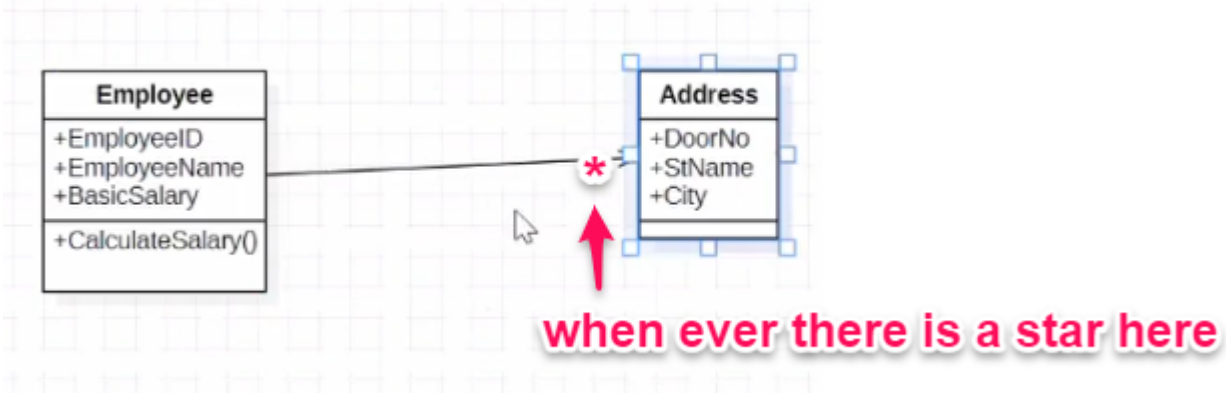
it is a list of address

```
//add address to employee
List<Address> addresss = new List<Address>();
addresss.Add(address);
employee.address = addresss;
```

so to add 1 object also i have to create a list
then add that address to the list
and then assign it back here
Don't you feel like this is follish code ?
so who develop the employee class it is his responsiblity that it stores 1 address or 100 address that is not my problem
why should i create a list
add this data to the list

and then assign list back here

So whenever you have to achieve many,



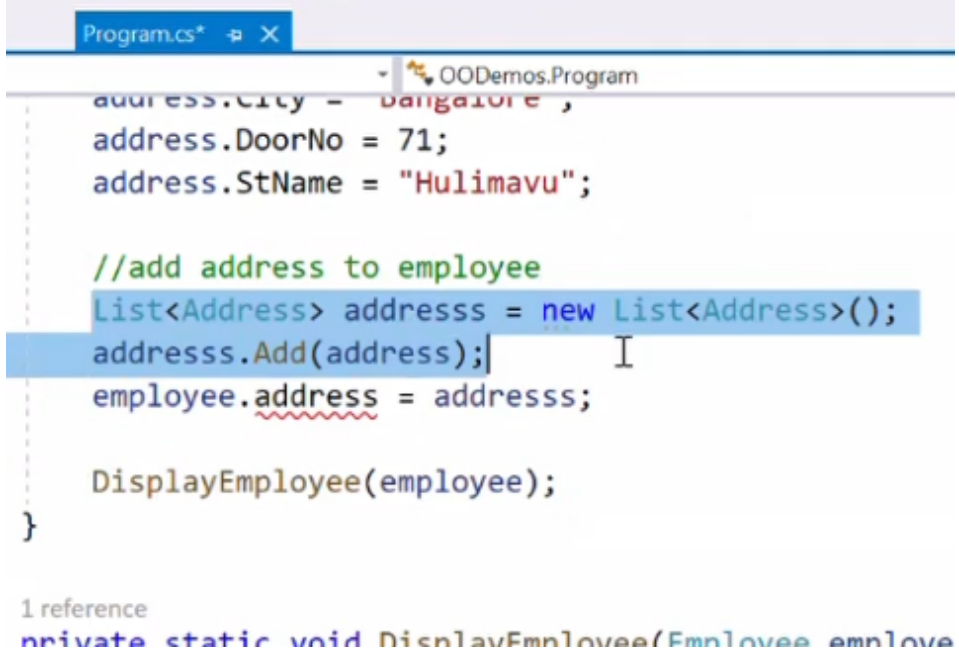
You should always create a private list

```
    2 references
    public int EmployeeId { get; set; }
    2 references
    public string EmployeeName { get; set; }

    private  List<Address> address = new List<Address>();

}
```

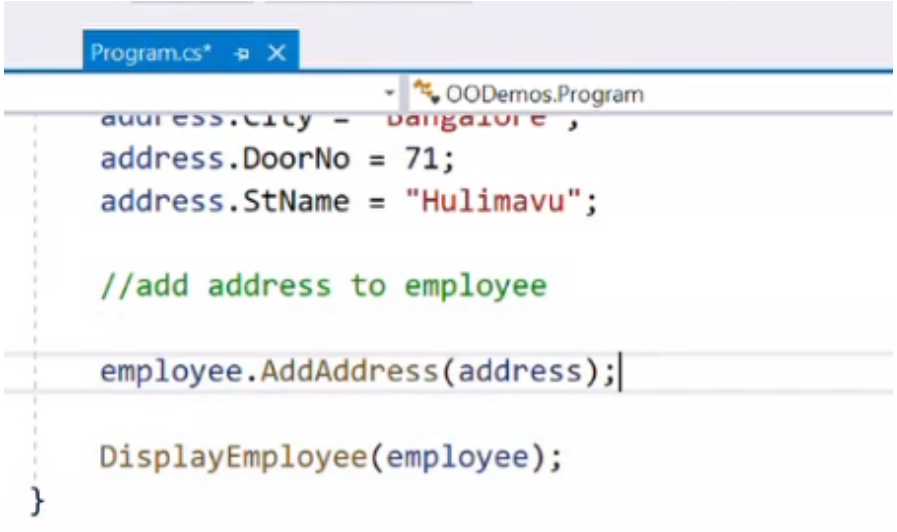
and create two public method AddAddress() which will take a address, GetAddress() which give me the list of address

So whenever you have list created, so list should always have 2 methods in it 1 to add the data 1 to get the data

So, what's the advantage other person will get, it is simple he will not bother about creating all these things



he will directly goto empoloyee.add method or employee.get method



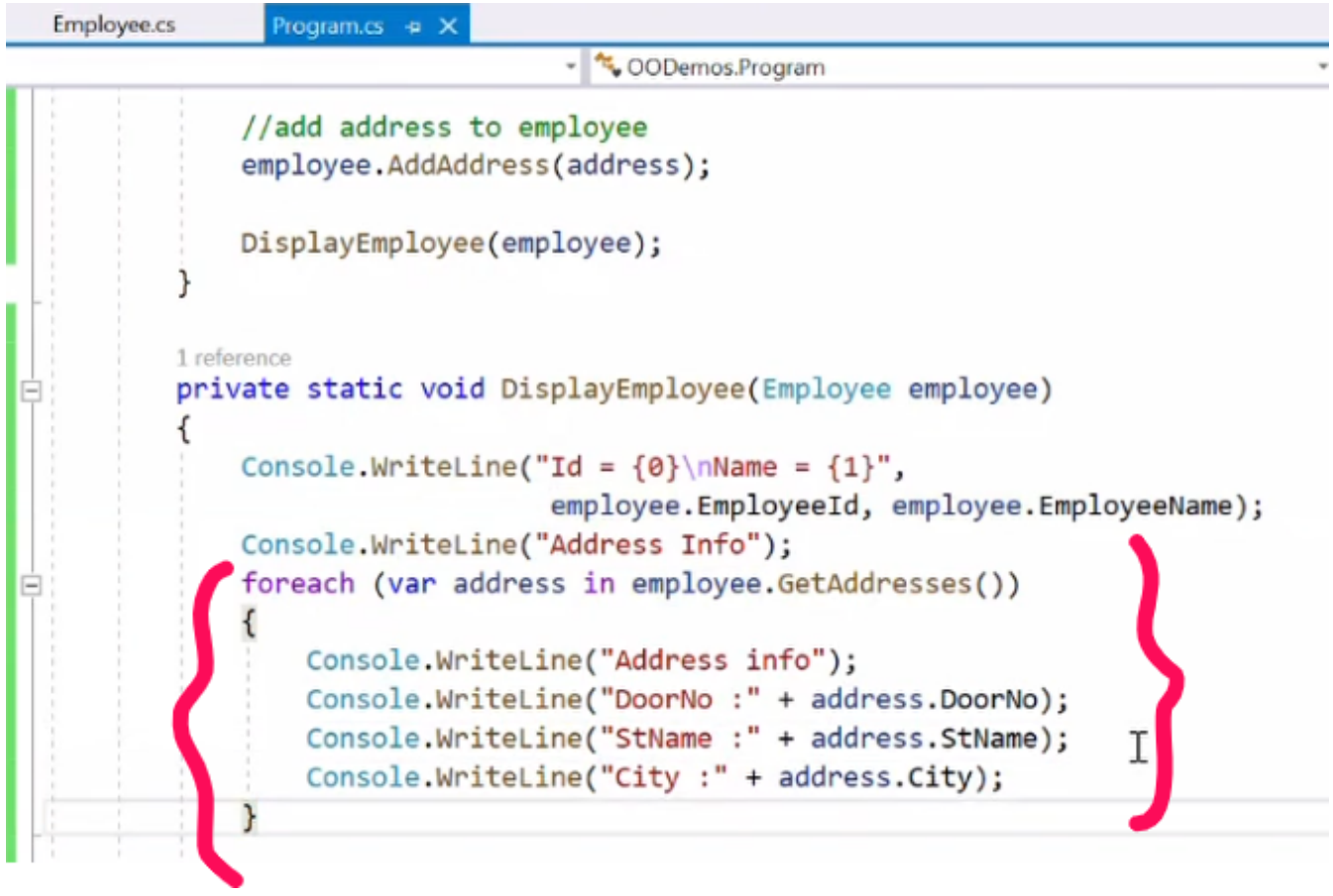
Now how will i access the address data ?

```
1 reference
private static void DisplayEmployee(Employee employee)
{
    Console.WriteLine("Id = {0}\nName = {1}",
                      employee.EmployeeId, employee.EmployeeName);

    Console.WriteLine("Address info");
    Console.WriteLine("DoorNo :" + employee.Address.DoorNo);
    Console.WriteLine("StName :" + employee.Address.StName);
    Console.WriteLine("City :" + employee.Address.City);
```

this is for one address
so employee has address, address has door no
employee has address, address has stname
employe has address, address has city

but 1 employee may have more than 1 address therefore to fetch that data we use foreach loop

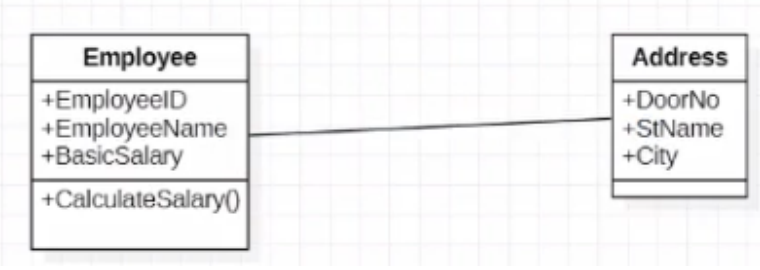
so employee has GetAddresses method and this can be more than one address **So i'll put a foreach loop here**



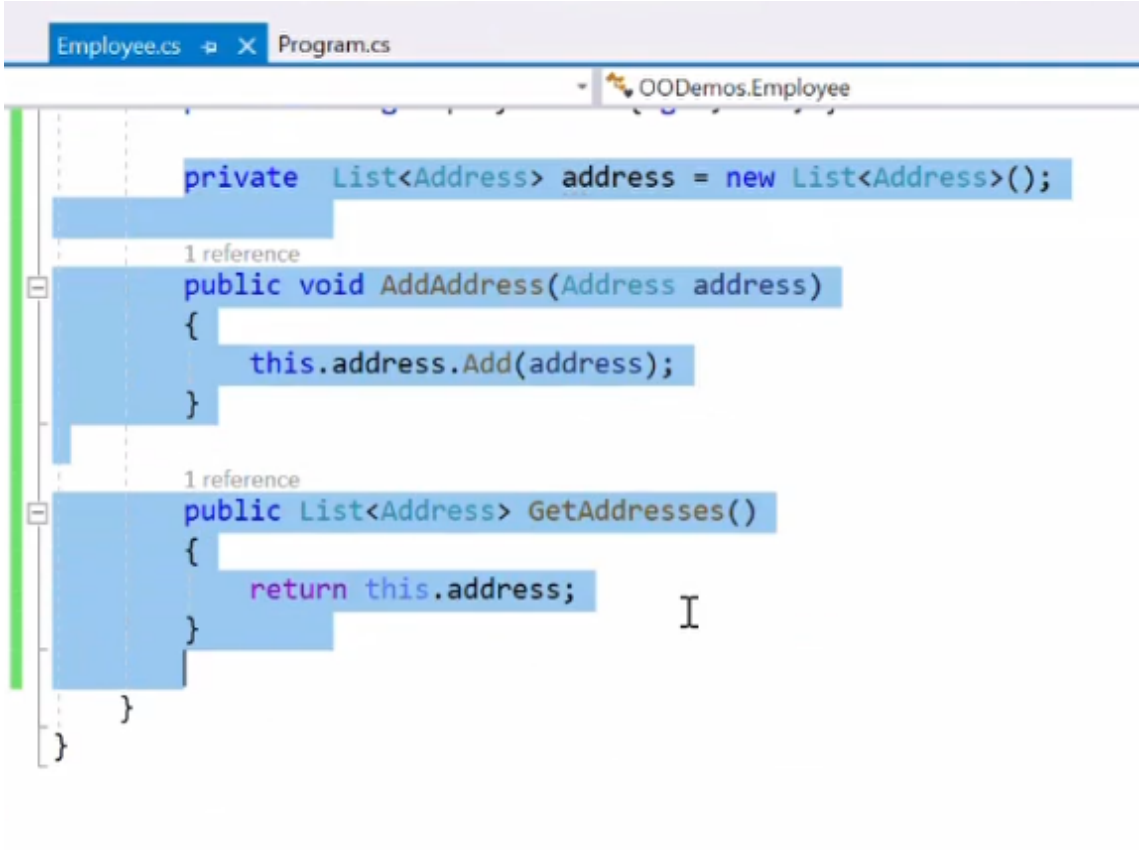
So many means i have to create list, I can create any data structure whatever we have in the liabrary but list is something very easy to handle that's a reason we are taking that as list

How to achieve BiDirection like employee has address and address has employee ?

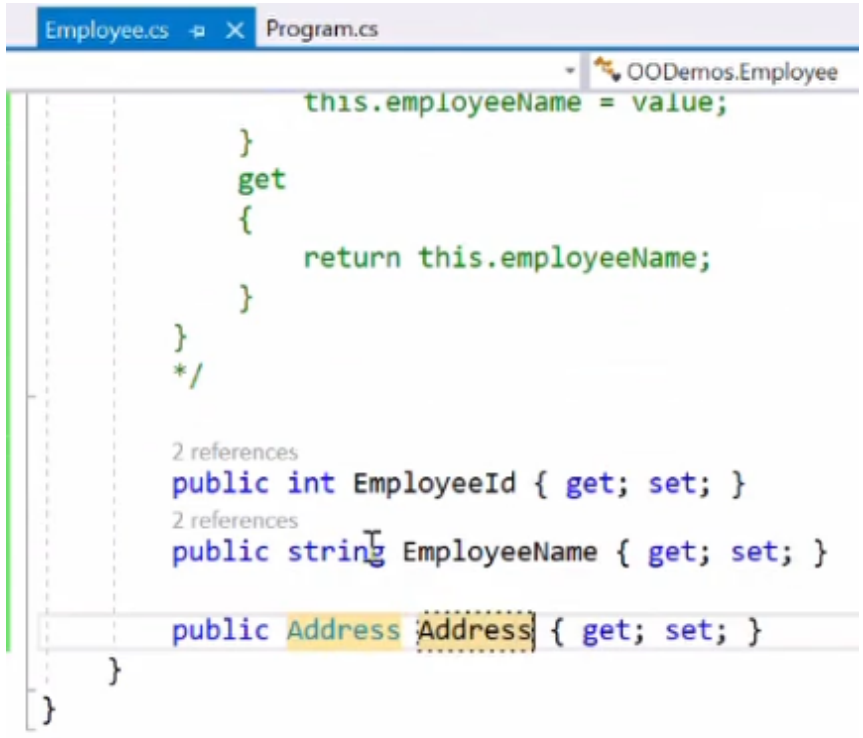
for now we are going with single object not list of object



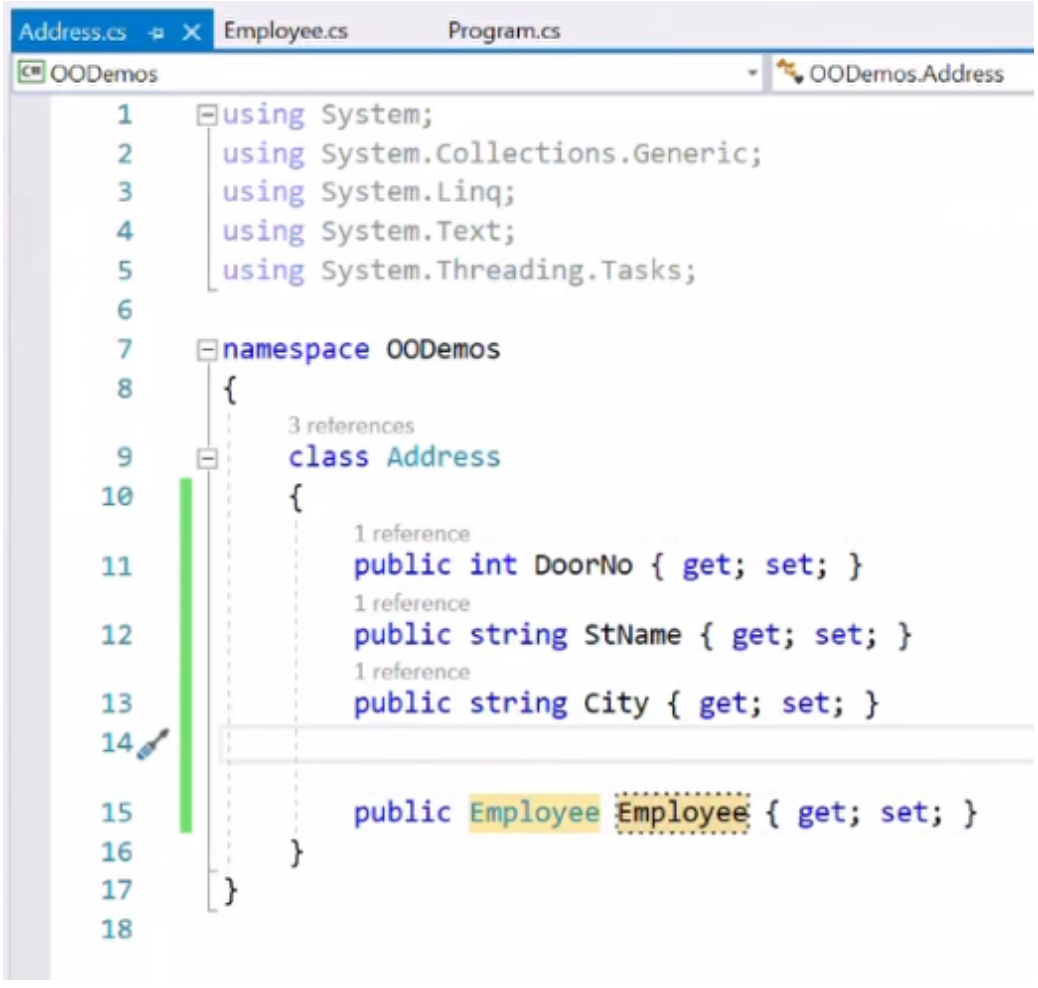
so i'll remove list part

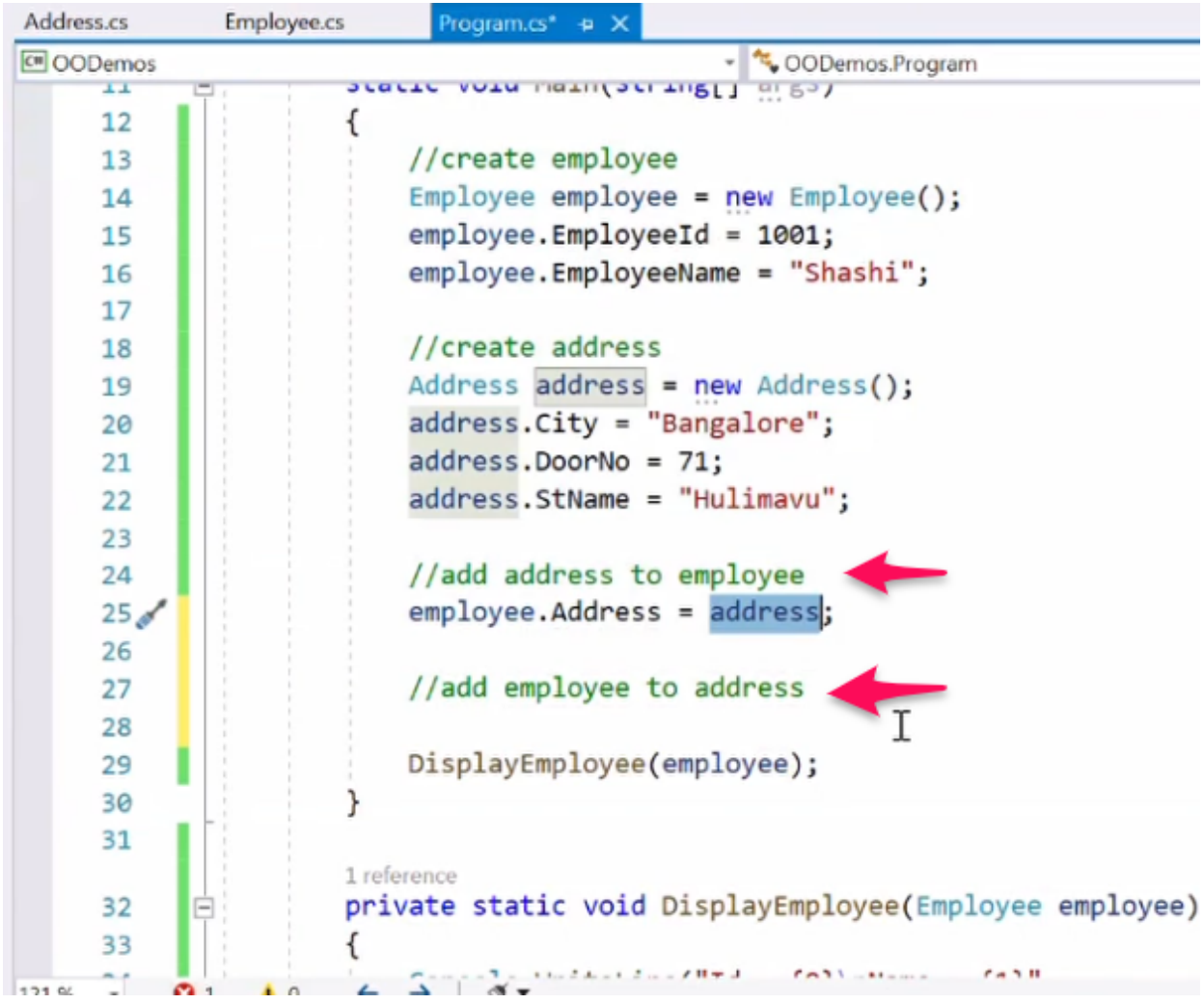


i'll say employee has a address



in the same way my address will also has the employee


Address.cs
Employee.cs
Program.cs
OODemos
OODemos.Address
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace OODemos
{
3 references
class Address
{
1 reference
public int DoorNo { get; set; }
1 reference
public string StName { get; set; }
1 reference
public string City { get; set; }
public Employee Employee { get; set; }
}
}



it is like i'm adding a order to a customer i'll say customer.order = order

now customer will have a order when you go to customer screen you can see customer has all these order
at the same time order will also has the customer details is'nt it

otherwise to which address the order gets delivered that becomes a big ?

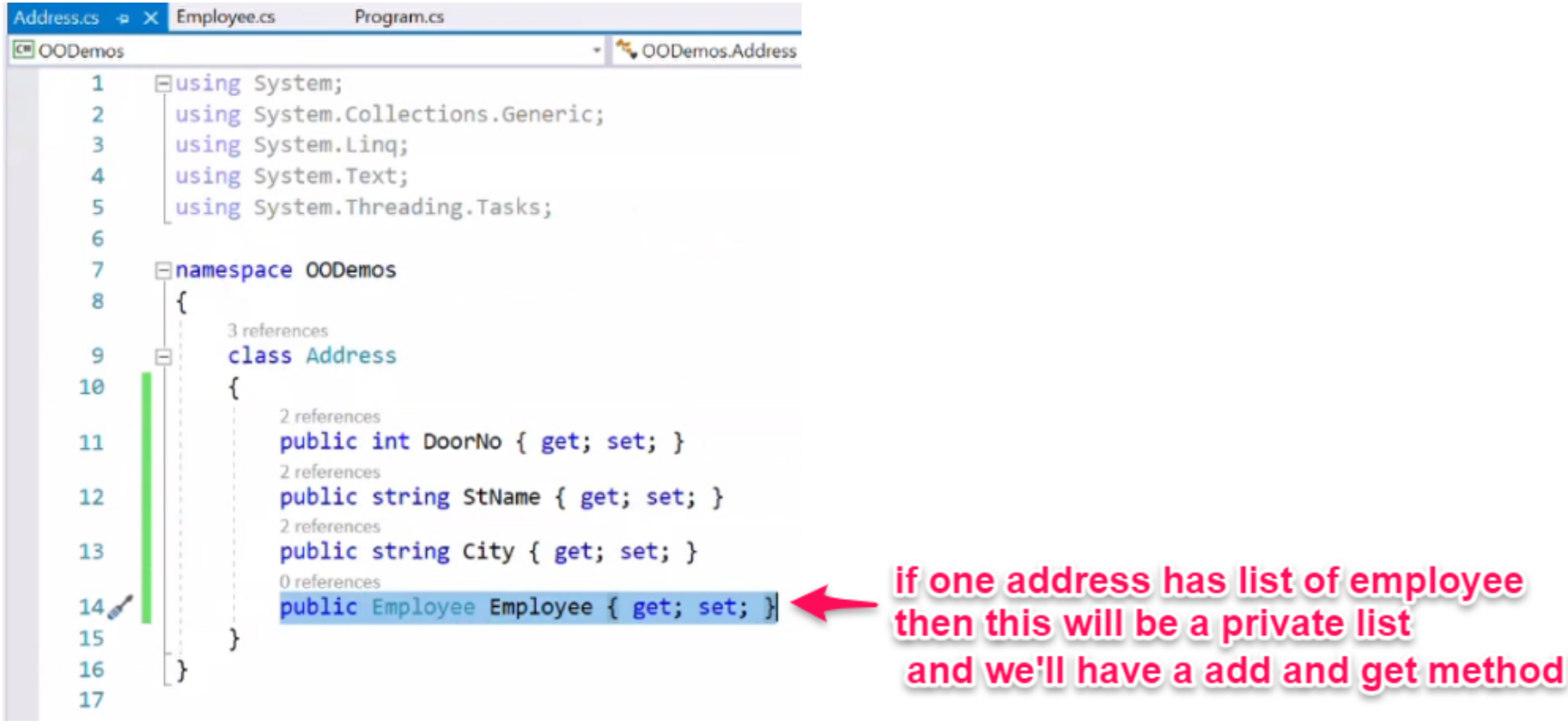
So how do you add a employee to the address
address.Employee = employee

```
//add address to employee
employee.Address = address;

//add employee to address
address.Employee = employee;
```

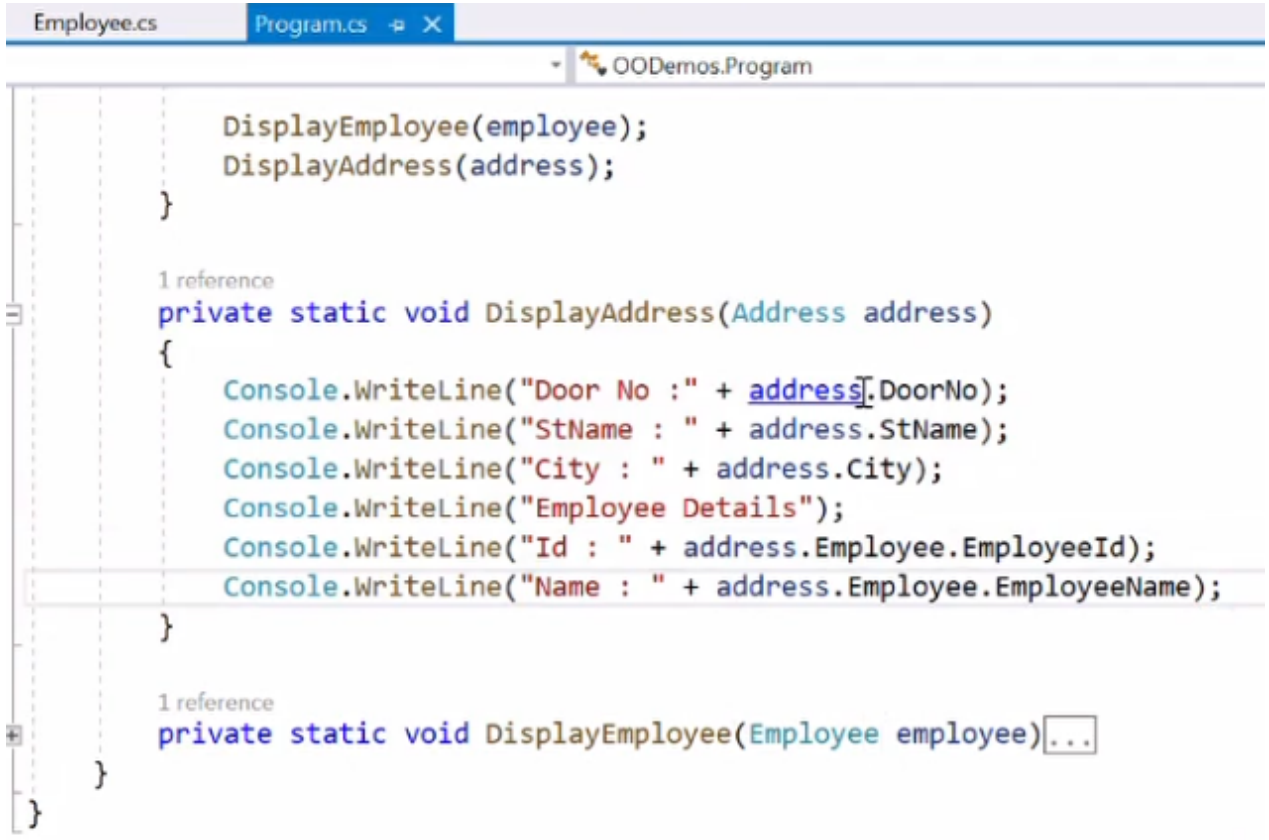
again this is how we fetch data for 1 employee

```
1 reference
private static void DisplayEmployee(Employee employee)
{
    Console.WriteLine("Id = {0}\nName = {1}",
                      employee.EmployeeId, employee.EmployeeName);
    Console.WriteLine("Address Info");
    Console.WriteLine("DoorNo :" + employee.Address.DoorNo);
    Console.WriteLine("StName :" + employee.Address.StName);
    Console.WriteLine("City :" + employee.Address.City);
```

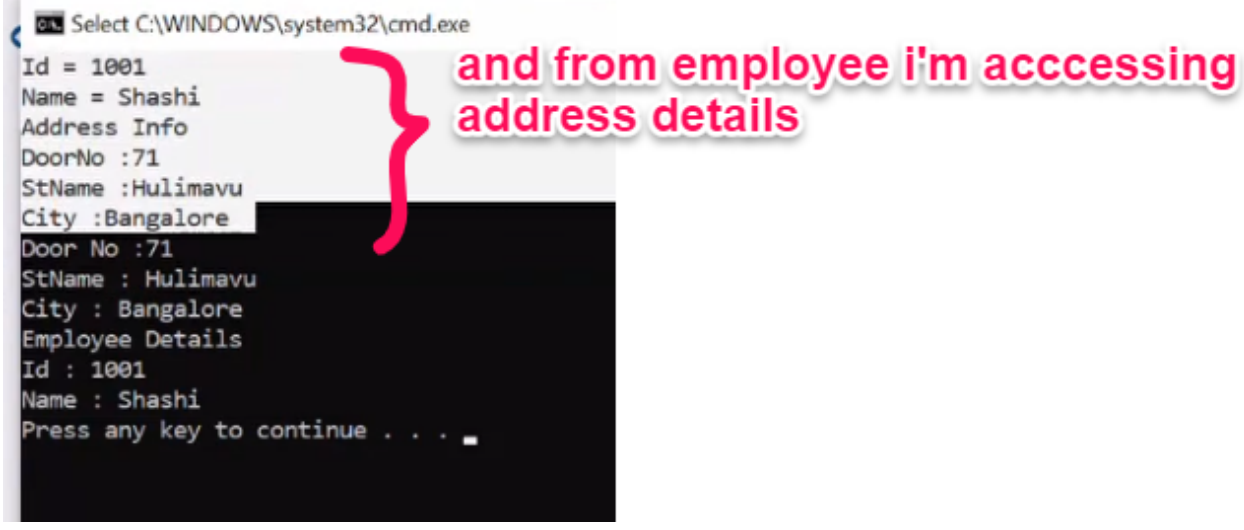
Now if a address has multiple employees then again you have to create a list of employees in address



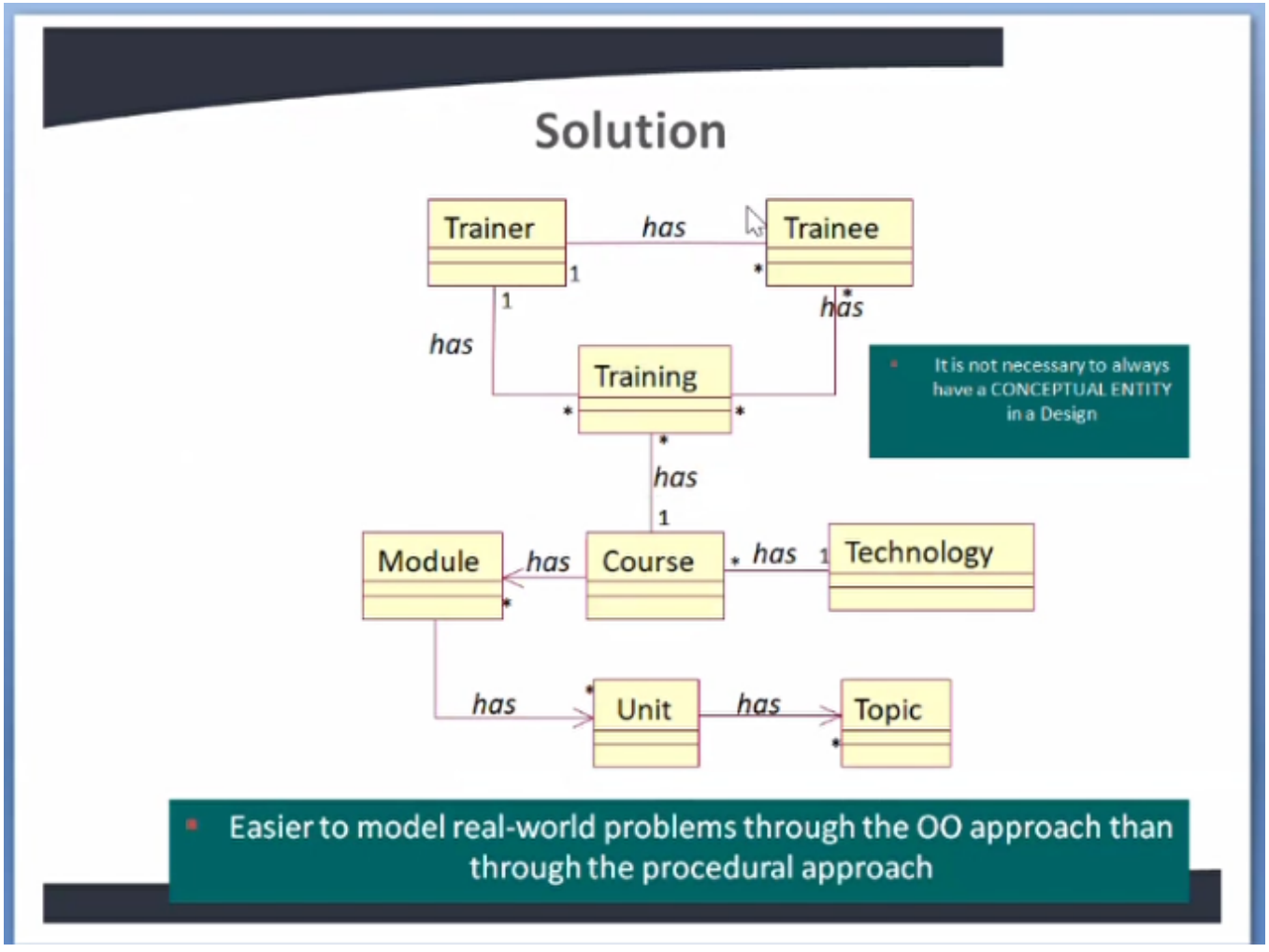
now the way i have display employee i'll say DisplayAddress();


Employee.cs
Program.cs
OODemos.Program
DisplayEmployee(employee);
DisplayAddress(address);
}
1 reference
private static void DisplayAddress(Address address)
{
Console.WriteLine("Door No :" + address.DoorNo);
Console.WriteLine("StName : " + address.StName);
Console.WriteLine("City : " + address.City);
Console.WriteLine("Employee Details");
Console.WriteLine("Id : " + address.Employee.EmployeeId);
Console.WriteLine("Name : " + address.Employee.EmployeeName);
}
1 reference
private static void DisplayEmployee(Employee employee)...
}
}


Select C:\WINDOWS\system32\cmd.exe
Id = 1001
Name = Shashi
Address Info
DoorNo :71
StName :Hulimavu
City :Bangalore
Door No :71
StName : Hulimavu
City : Bangalore
Employee Details
Id : 1001
Name : Shashi
Press any key to continue . . .
look at this from the address i'm accessing the employee details

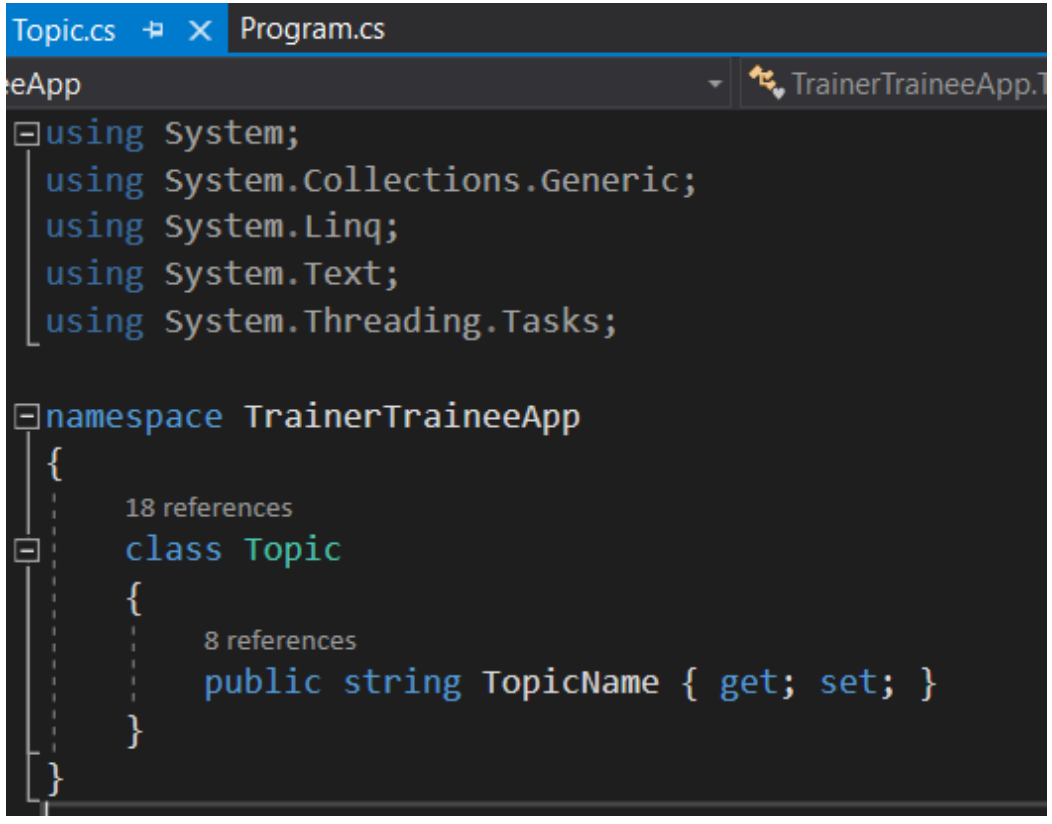

Select C:\WINDOWS\system32\cmd.exe
Id = 1001
Name = Shashi
Address Info
DoorNo :71
StName :Hulimavu
City :Bangalore
Door No :71
StName : Hulimavu
City : Bangalore
Employee Details
Id : 1001
Name : Shashi
Press any key to continue . . .
and from employee i'm acccessing address details

now lets code Trainer trainee

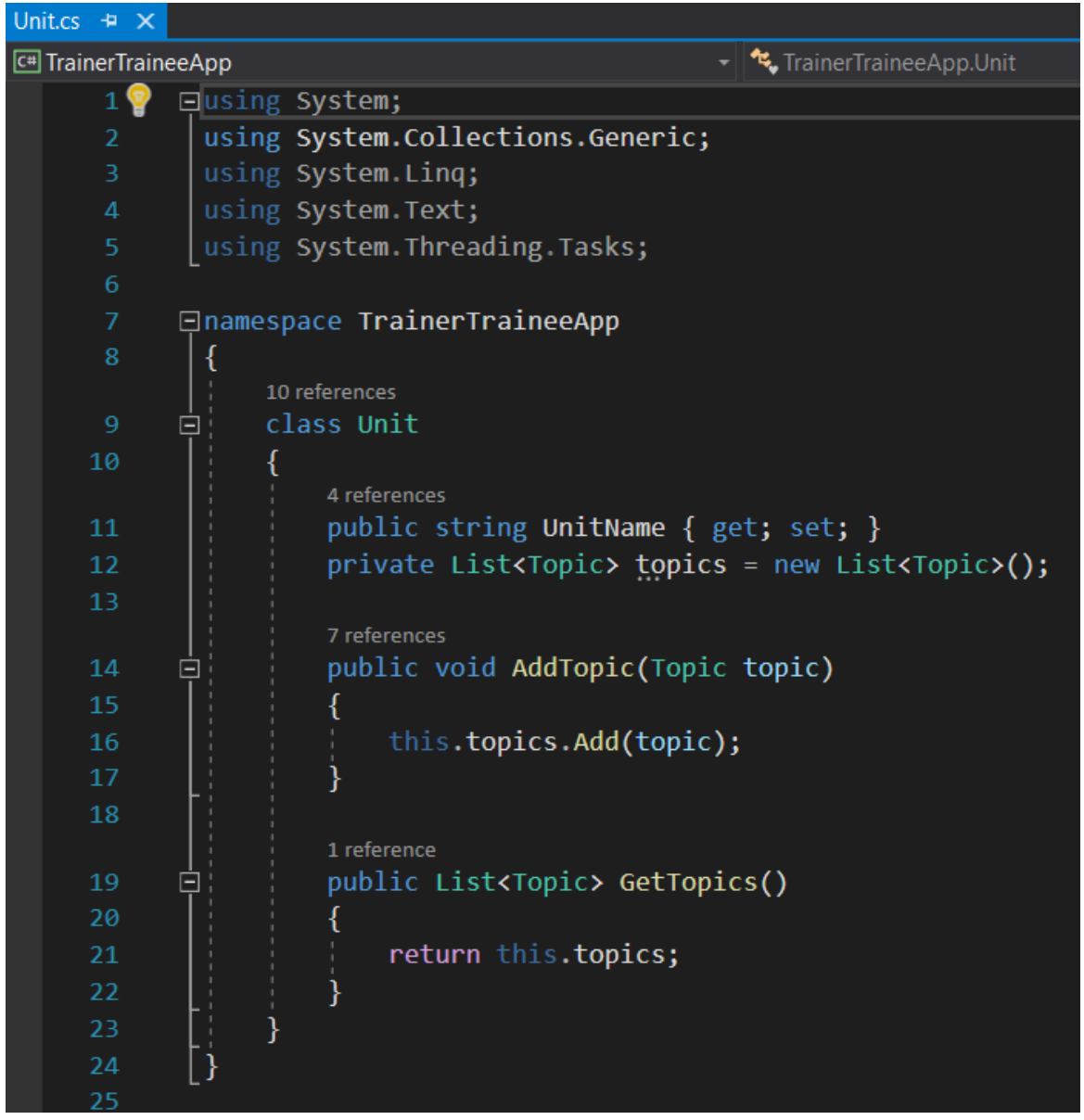


we can start coding from any where but we start with class which dose'nt have any dependency so topic and technology will not have any dependency so we'll go with topic

so we create topic class having property TopicName **Remember property will always have pascal casing**

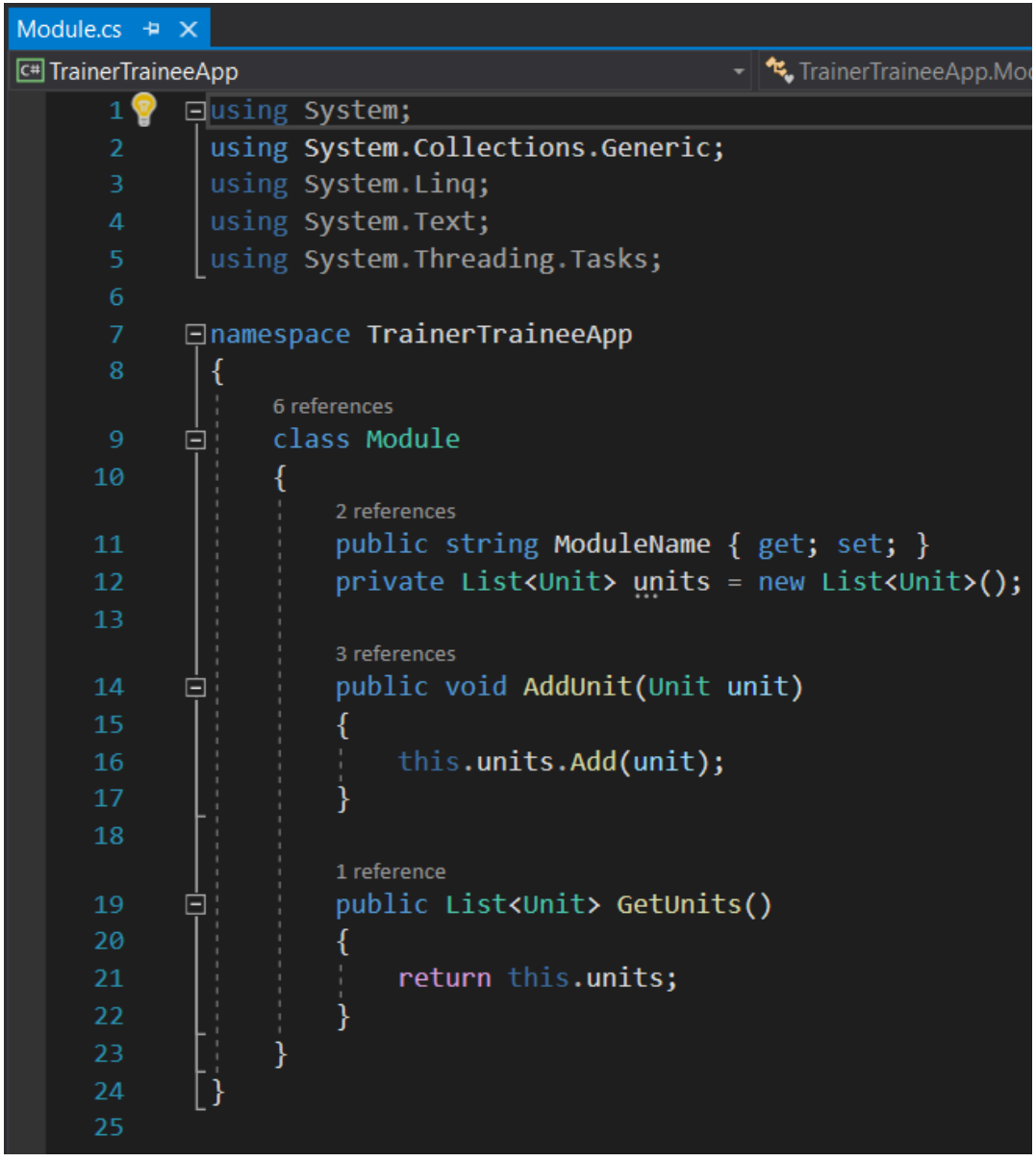


next is unit, unit will have UnitName and topic, How many topic? 1, 2, 10 .....
list of topics


Unit.cs
TrainerTraineeApp
TrainerTraineeApp.Unit
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace TrainerTraineeApp
{
10 references
class Unit
{
4 references
public string UnitName { get; set; }
private List<Topic> topics = new List<Topic>();
7 references
public void AddTopic(Topic topic)
{
this.topics.Add(topic);
}
1 reference
public List<Topic> GetTopics()
{
return this.topics;
}
}
}

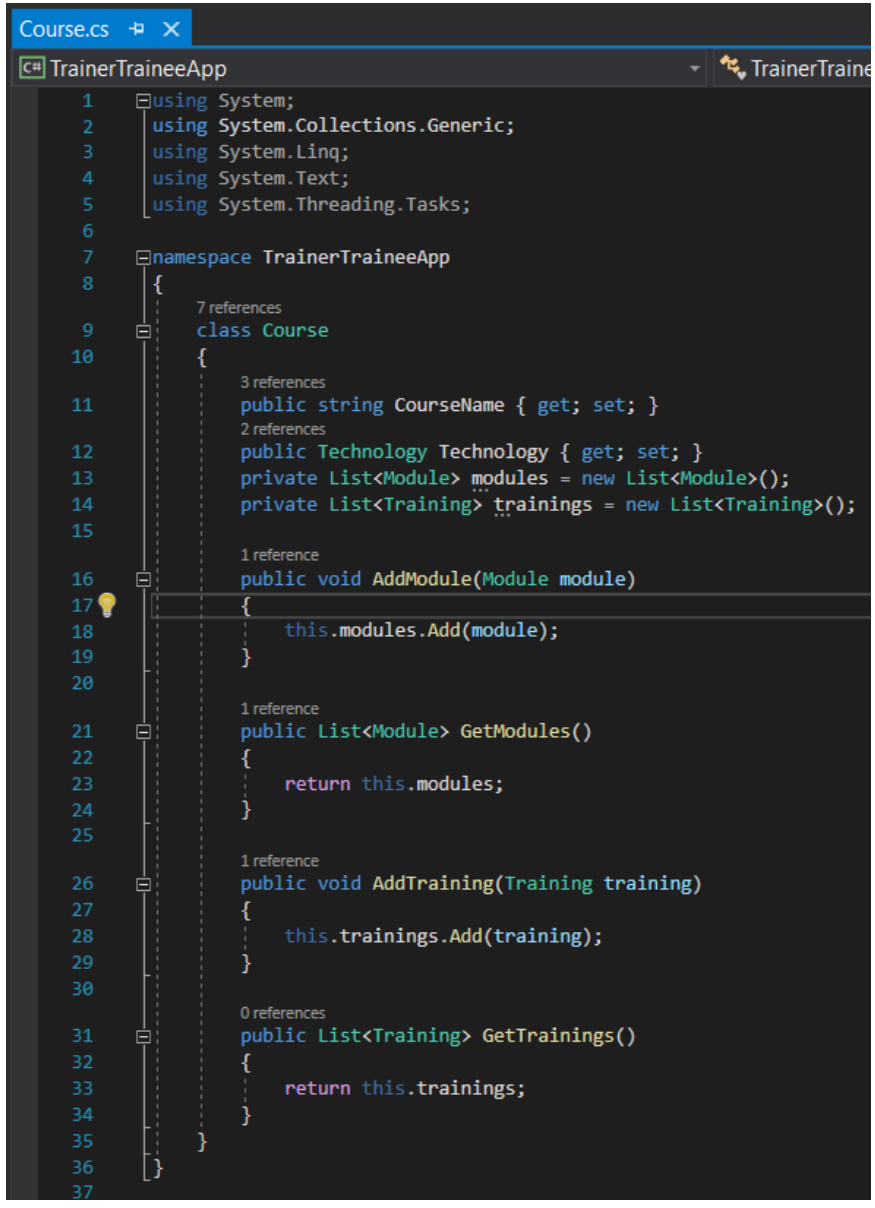
Now we create Module,

Module will have ModuleName, list of Units

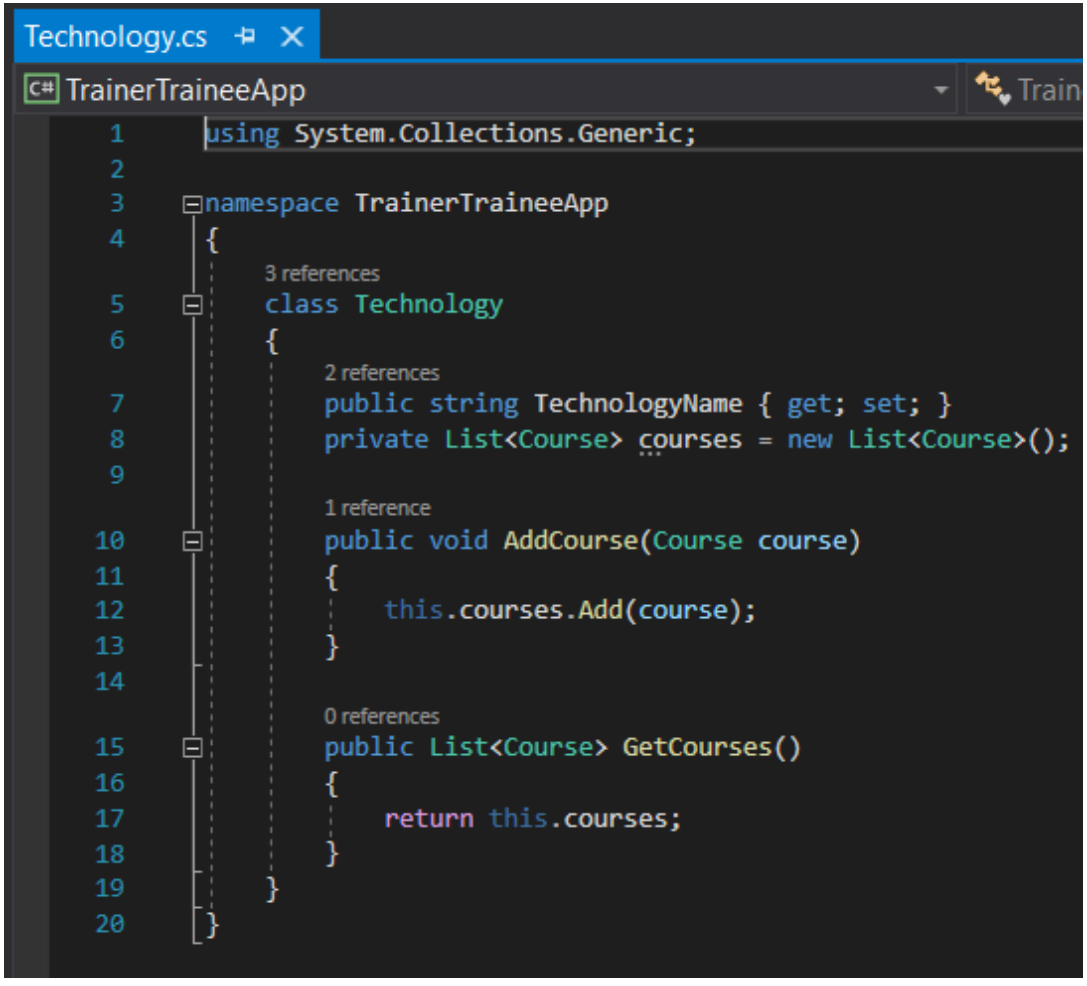

Module.cs
TrainerTraineeApp
TrainerTraineeApp.Mo
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace TrainerTraineeApp
{
6 references
class Module
{
2 references
public string ModuleName { get; set; }
private List<Unit> units = new List<Unit>();
3 references
public void AddUnit(Unit unit)
{
this.units.Add(unit);
}
1 reference
public List<Unit> GetUnits()
{
return this.units;
}
}
}

next we create Course

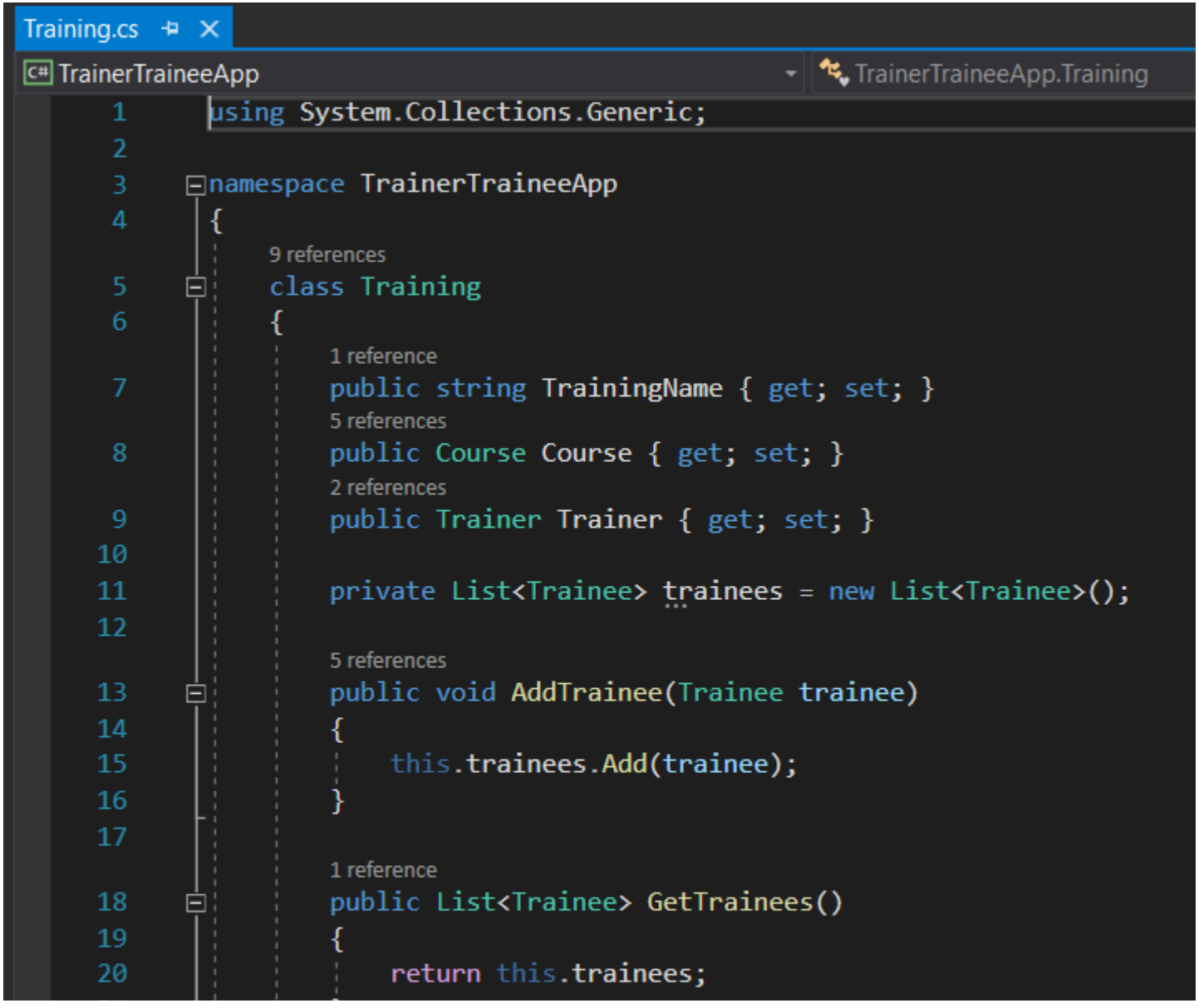
Course will have CourseName, technology, list of modules, list of trainings


Course.cs
TrainerTraineeApp
TrainerTraine
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace TrainerTraineeApp
{
7 references
class Course
{
3 references
public string CourseName { get; set; }
2 references
public Technology Technology { get; set; }
private List<Module> modules = new List<Module>();
private List<Training> trainings = new List<Training>();
1 reference
public void AddModule(Module module)
{
this.modules.Add(module);
}
1 reference
public List<Module> GetModules()
{
return this.modules;
}
1 reference
public void AddTraining(Training training)
{
this.trainings.Add(training);
}
0 references
public List<Training> GetTrainings()
{
return this.trainings;
}
}
}

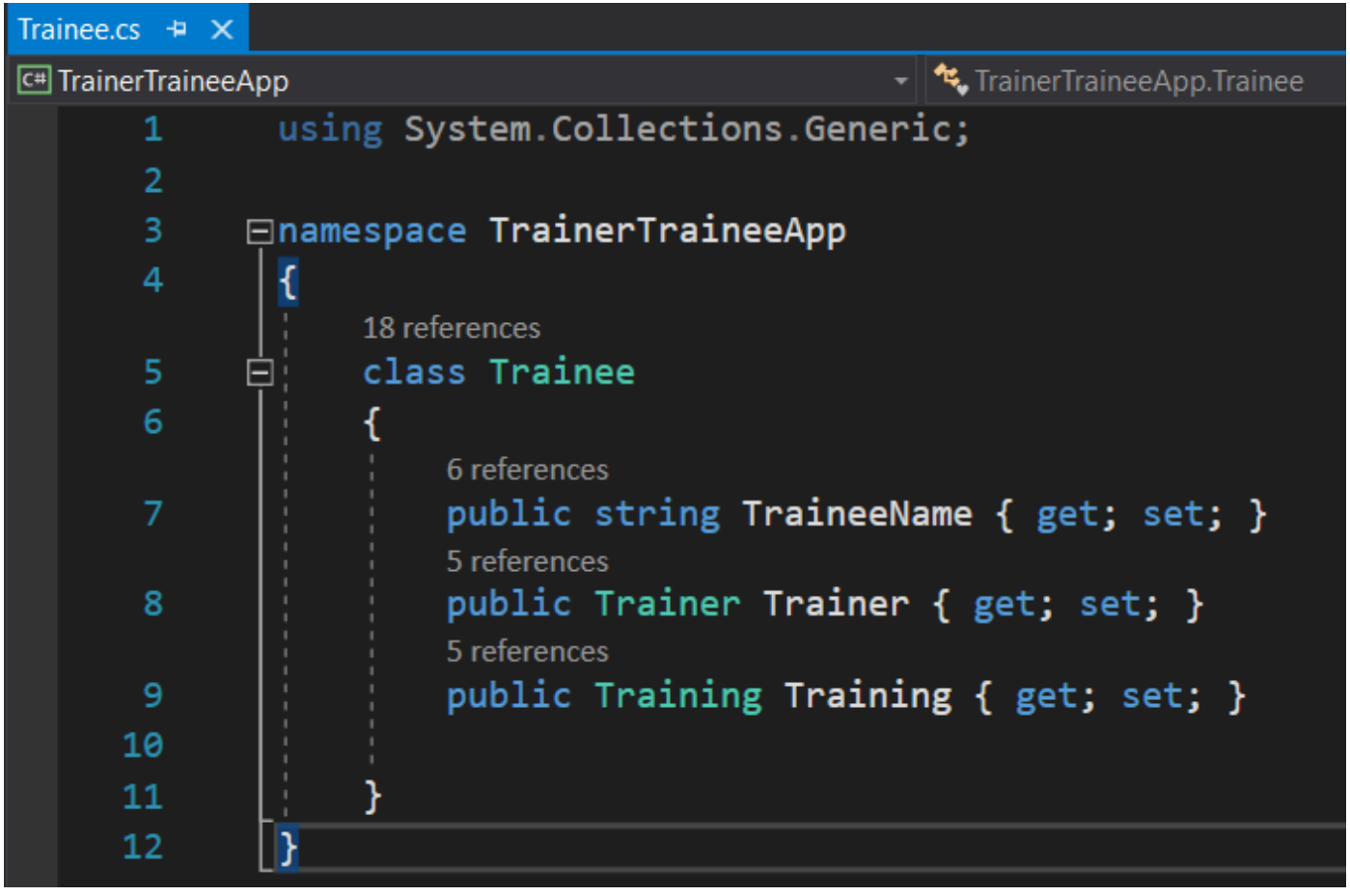
then we create technology, in technology we have TechnologyName and Course



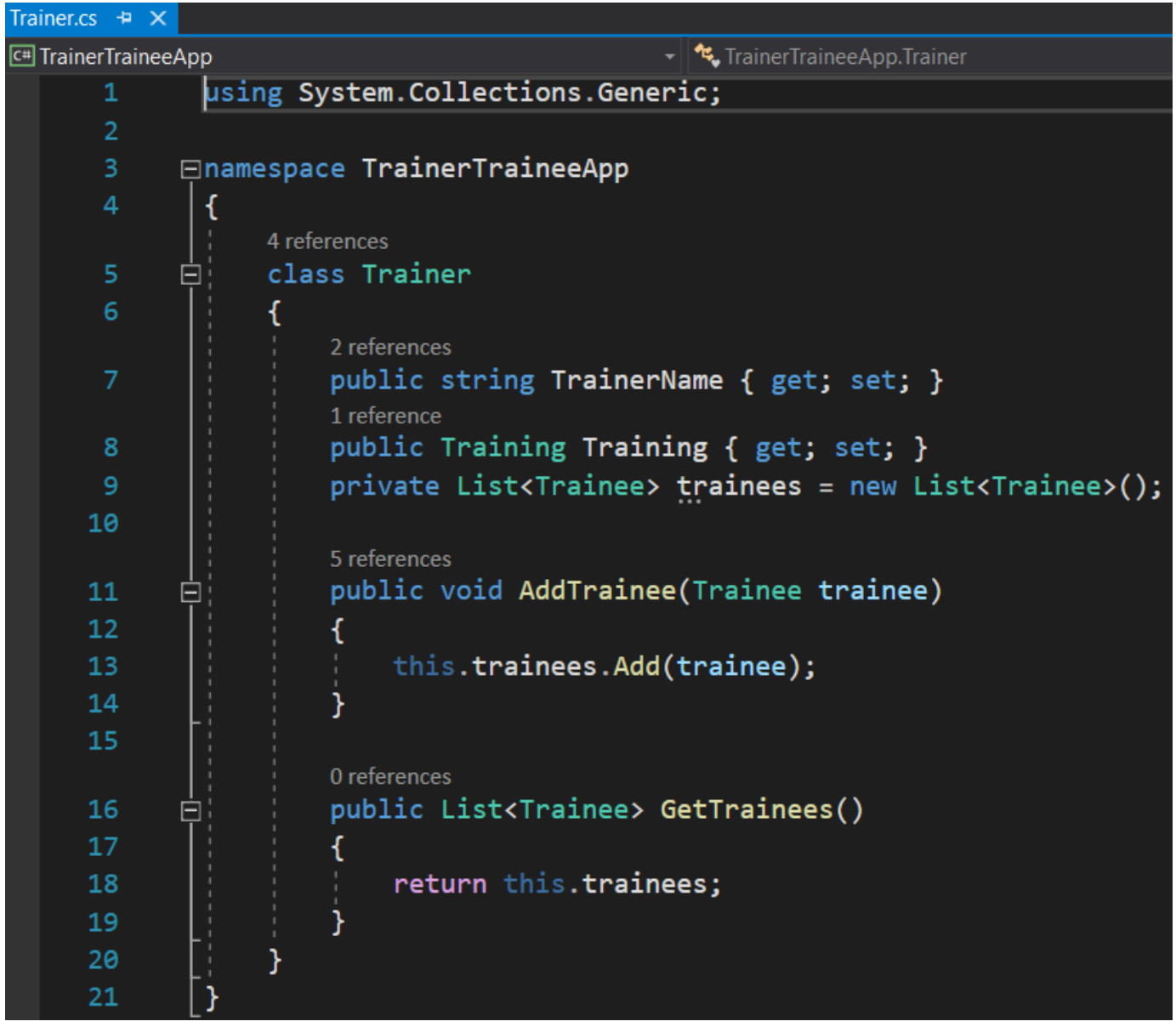
Now Training, in training we have TrainingName, Course, training has a trainer, and list or trainees


Training.cs
TrainerTraineeApp
TrainerTraineeApp.Training
using System.Collections.Generic;
namespace TrainerTraineeApp
{
9 references
class Training
{
1 reference
public string TrainingName { get; set; }
5 references
public Course Course { get; set; }
2 references
public Trainer Trainer { get; set; }
private List<Trainee> trainees = new List<Trainee>();
5 references
public void AddTrainee(Trainee trainee)
{
this.trainees.Add(trainee);
}
1 reference
public List<Trainee> GetTrainees()
{
return this.trainees;

Now we have trainee, trainee has TraineeName, Trainer, and Trainer **here there is a star but at any given point of time we are concentrating on one training thats why we have one training otherwise it should be list of training**



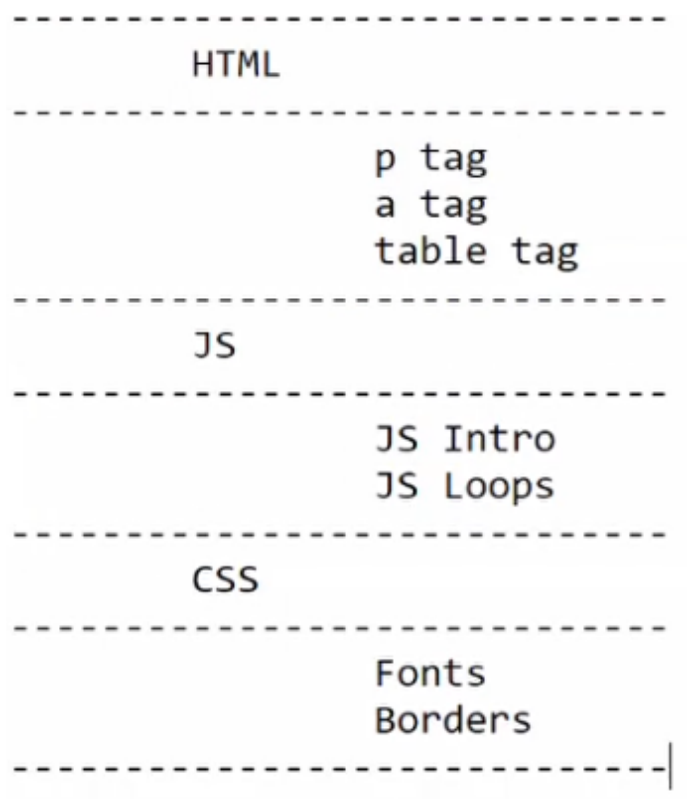
Now Trainer


Trainer.cs
TrainerTraineeApp
TrainerTraineeApp.Trainer
using System.Collections.Generic;
namespace TrainerTraineeApp
{
4 references
class Trainer
{
2 references
public string TrainerName { get; set; }
1 reference
public Training Training { get; set; }
private List<Trainee> trainees = new List<Trainee>();
5 references
public void AddTrainee(Trainee trainee)
{
this.trainees.Add(trainee);
}
0 references
public List<Trainee> GetTrainees()
{
return this.trainees;
}
}
}

Now our model classes is ready, So i want to print something like this on the console

File Edit Format View Help

```
        Training Details
-------------------------------------
Course: .NET
Training Name: MP-.NET 202
Trainer: Shashikanth

Trainees List
-------------
x
x
x
x
x

Course Details  - Technology: x
------------------------------
Web Fun
------------------------------
        HTML
------------------------------
                p tag
```



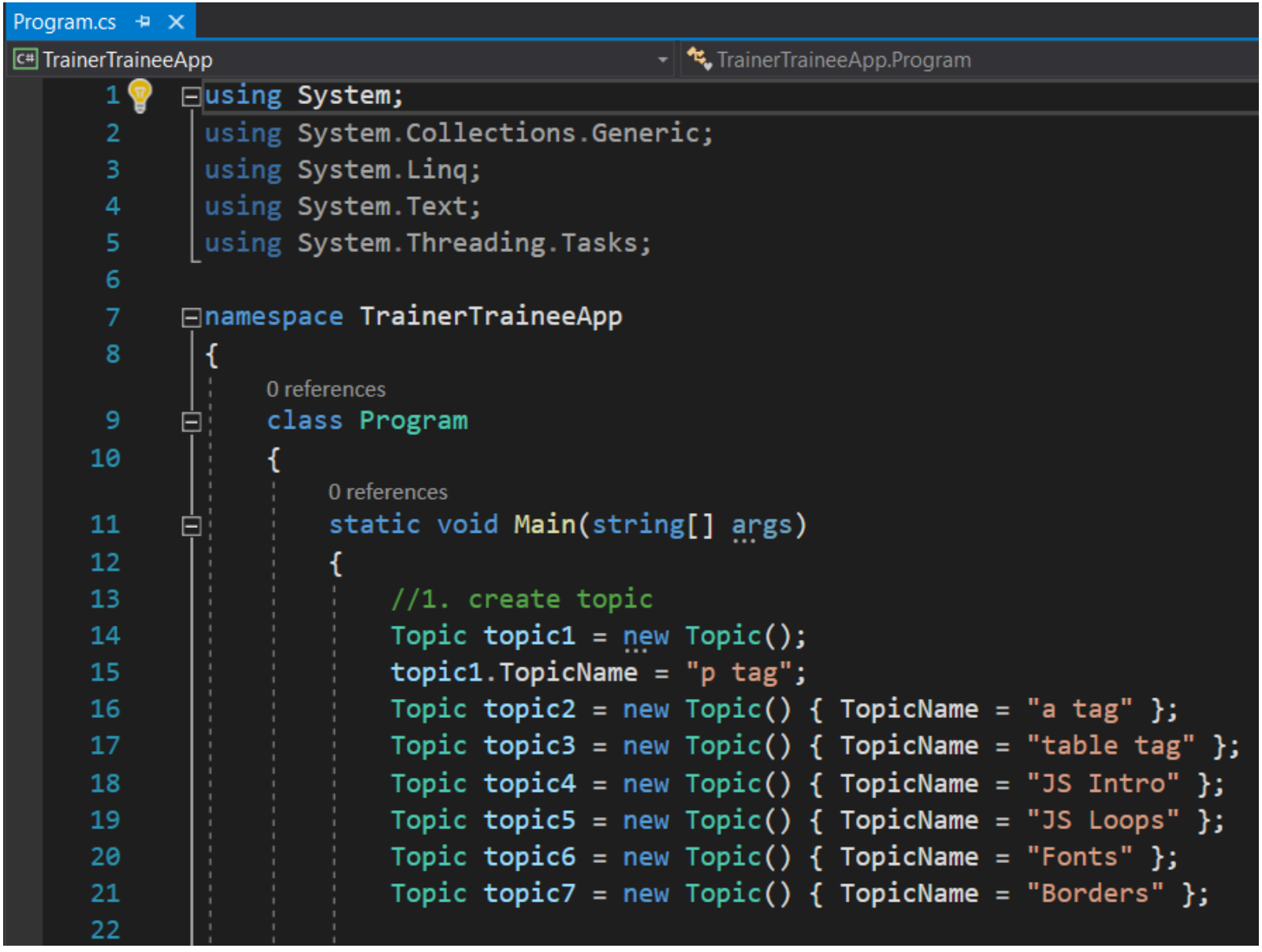
this should be the output

so 1. you have to create topic

2. create unit
3. add topic to unit
4. create module
5. add unit to module

6. create course
7. add module to course
8. create technology
9. add technology to course
10. add course to technology
11. create training
12. add course to training
13. add training to course
14. create trainee
15. add training to trainee
16. add trainee to training
17. create trainer
18. add training to trainer
19. add trainer to training
20. add trainee to trainer
21. add trainer to trainee
22. display training details

So lets starts


Program.cs
TrainerTraineeApp
TrainerTraineeApp.Program
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace TrainerTraineeApp
{
0 references
class Program
{
0 references
static void Main(string[] args)
{
//1. create topic
Topic topic1 = new Topic();
topic1.TopicName = "p tag";
Topic topic2 = new Topic() { TopicName = "a tag" };
Topic topic3 = new Topic() { TopicName = "table tag" };
Topic topic4 = new Topic() { TopicName = "JS Intro" };
Topic topic5 = new Topic() { TopicName = "JS Loops" };
Topic topic6 = new Topic() { TopicName = "Fonts" };
Topic topic7 = new Topic() { TopicName = "Borders" };

```
//2. create unit
Unit unit1 = new Unit() { UnitName = "HTML" };
Unit unit2 = new Unit() { UnitName = "JS" };
Unit unit3 = new Unit() { UnitName = "CSS" };
```

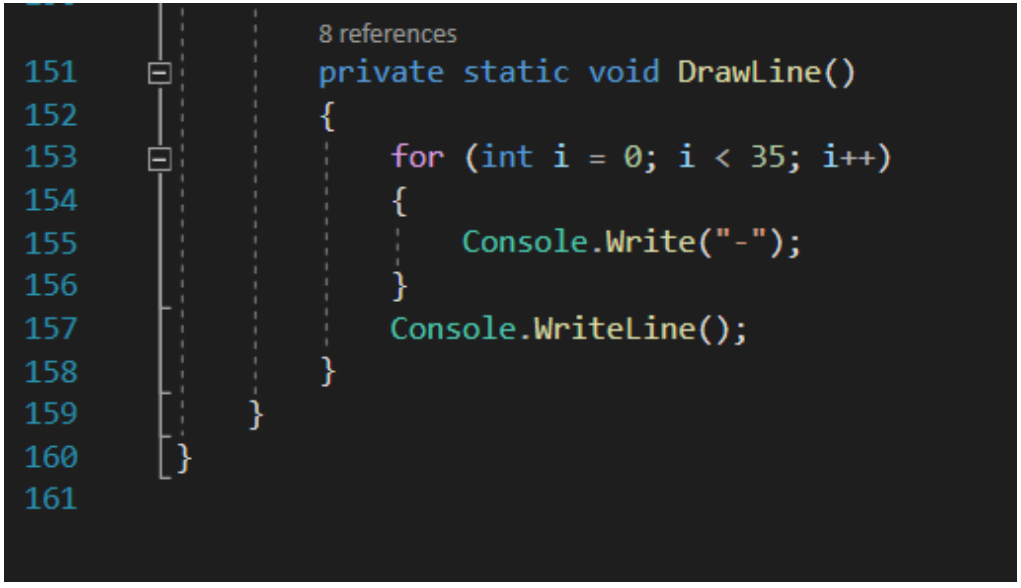
```
//3. add topic to unit
unit1.AddTopic(topic1);
unit1.AddTopic(topic2);
unit1.AddTopic(topic3);
unit2.AddTopic(topic4);
unit2.AddTopic(topic5);
unit3.AddTopic(topic6);
unit3.AddTopic(topic7);
```

```
//4. create module
Module module1 = new Module() { ModuleName = "Web Fun" };

//5. add unit to module
module1.AddUnit(unit1);
module1.AddUnit(unit2);
module1.AddUnit(unit3);

//6. create course
Course course = new Course() { CourseName = ".NET" };

//7. add moudle to course
course.AddModule(module1);

//8. create technology
Technology technology = new Technology() { TechnologyName = ".NET Technology" };

//9. add technology to course
course.Technology = technology;

//10. add course to technology
technology.AddCourse(course);

//11. create training
Training training = new Training() { TrainingName = "MP-.NET 202" };
```

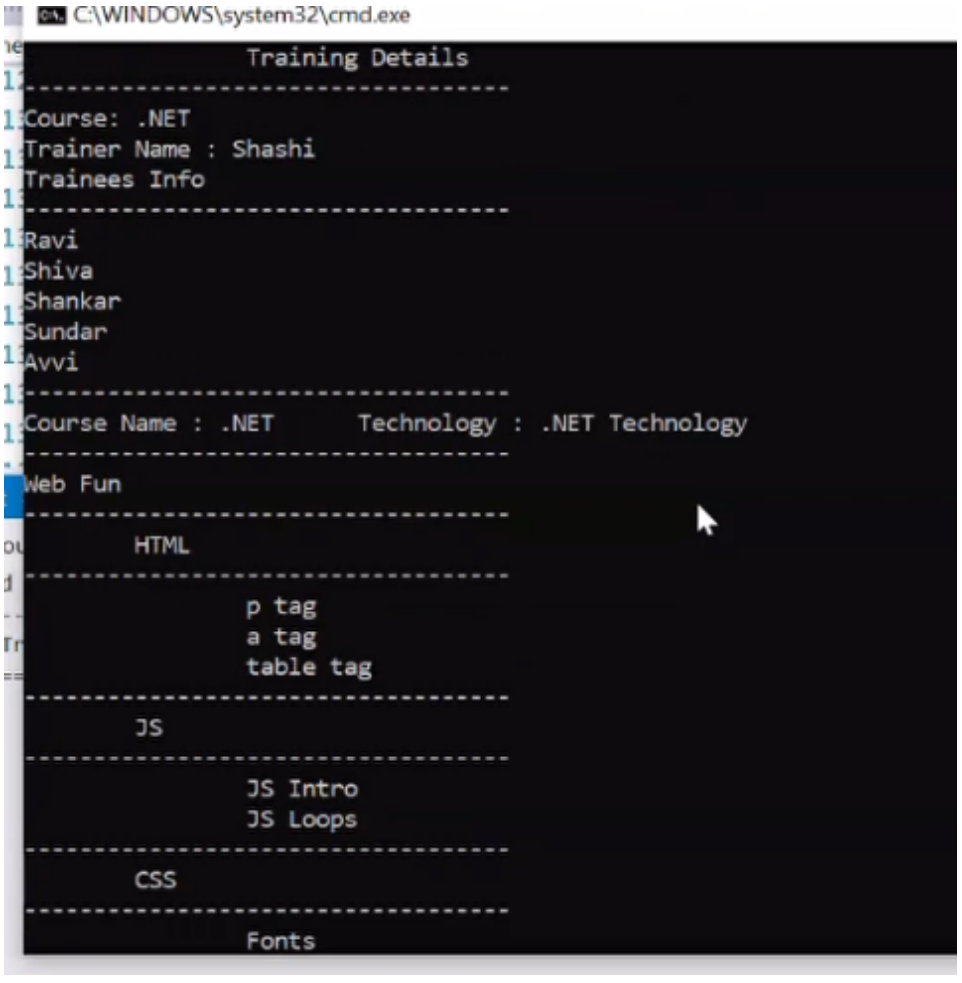
```
//12. add course to training
training.Course = course;

//13. add training to course
course.AddTraining(training);

//14. create trainee
Trainee trainee1 = new Trainee() { TraineeName = "Ravi" };
Trainee trainee2 = new Trainee() { TraineeName = "Shiva" };
Trainee trainee3 = new Trainee() { TraineeName = "Shankar" };
Trainee trainee4 = new Trainee() { TraineeName = "Sundar" };
Trainee trainee5 = new Trainee() { TraineeName = "Avvi" };

//15. add training to trainee
trainee1.Training = training;
trainee2.Training = training;
trainee3.Training = training;
trainee4.Training = training;
trainee5.Training = training;

//16. add trainee to training
training.AddTrainee(trainee1);
training.AddTrainee(trainee2);
training.AddTrainee(trainee3);
training.AddTrainee(trainee4);
training.AddTrainee(trainee5);
```

```
    //17. create trainer
    Trainer trainer = new Trainer() { TrainerName = "Shashi" };

    //18. add training to trainer
    trainer.Training = training;

    //19. add trainer to training
    training.Trainer = trainer;

    //20. add trainee to trainer
    trainer.AddTrainee(trainee1);
    trainer.AddTrainee(trainee2);
    trainer.AddTrainee(trainee3);
    trainer.AddTrainee(trainee4);
    trainer.AddTrainee(trainee5);

    //21. add trainer to trainee
    trainee1.Trainer = trainer;
    trainee2.Trainer = trainer;
    trainee3.Trainer = trainer;
    trainee4.Trainer = trainer;
    trainee5.Trainer = trainer;

    //22. dispaly training details
    DisplayTraining(training);
}
```

```
1 reference
private static void DisplayTraining(Training training)
{
    Console.WriteLine("\t\tTraining Details");
    DrawLine();
    Console.WriteLine("Course: " + training.Course.CourseName);
    Console.WriteLine("Trainer Name : " + training.Trainer.TrainerName);
    Console.WriteLine("Trainees Info");
    DrawLine();
    foreach (var trainee in training.GetTrainees())
    {
        Console.WriteLine(trainee.TraineeName);
    }
    DrawLine();
    Console.WriteLine("Course Name : {0}\tTechnology : {1}",
                      training.Course.CourseName, training.Course.Technology.TechnologyName);
    DrawLine();
    foreach (var module in training.Course.GetModules())
    {
        Console.WriteLine(module.ModuleName);
        DrawLine();
        foreach (var unit in module.GetUnits())
        {
            Console.WriteLine("\t" + unit.UnitName);
            DrawLine();
            foreach (var topic in unit.GetTopics())
            {
                Console.WriteLine("\t\t" + topic.TopicName);
```


8 references
private static void DrawLine()
{
for (int i = 0; i < 35; i++)
{
Console.Write("-");
}
Console.WriteLine();
}
}
}

output


C:\WINDOWS\system32\cmd.exe
Training Details
Course: .NET
Trainer Name : Shashi
Trainees Info
Ravi
Shiva
Shankar
Sundar
Avvi
Course Name : .NET Technology : .NET Technology
Web Fun
HTML
p tag
a tag
table tag
JS
JS Intro
JS Loops
CSS
Fonts

Exercise

A company sells different items to customers who have placed orders. An order can be placed for several items. However, a company gives special discounts to its registered customers.

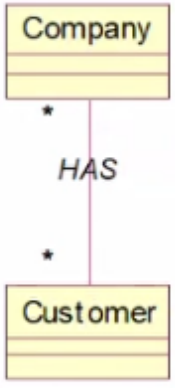
- Identify the different classes from the above problem statement
- Identify the different connections (relationships) between the above classes

Identifying Classes

- Company
- Item
- Order
- Customer
- RegCustomer

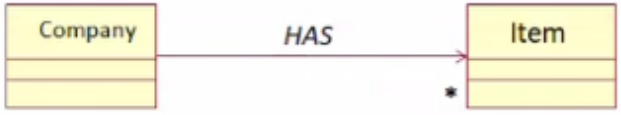
Identifying Relationships

- Company - Customer
  - Company 'HAS' many Customers
  - Customer 'HAS' many Companies



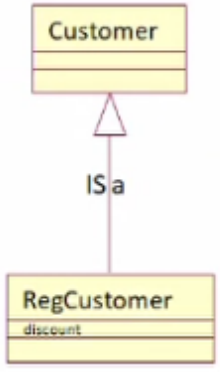
Identifying Relationships

- Company - Item
  - Company HAS many Items



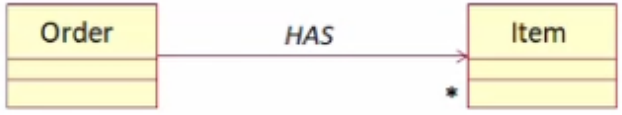
Identifying Relationships

- Customer - RegCustomer
  - RegCustomer 'IS' a Customer



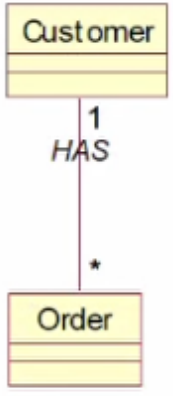
Identifying Relationships

- Order - Item
  - Order HAS many Items

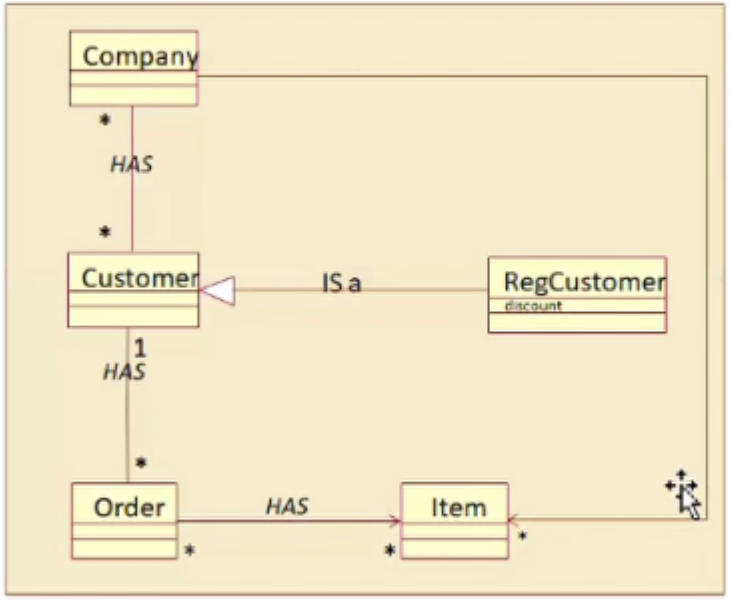


Identifying Relationships

- Customer - Order
  - Customer HAS many Orders
  - Order HAS one Customer

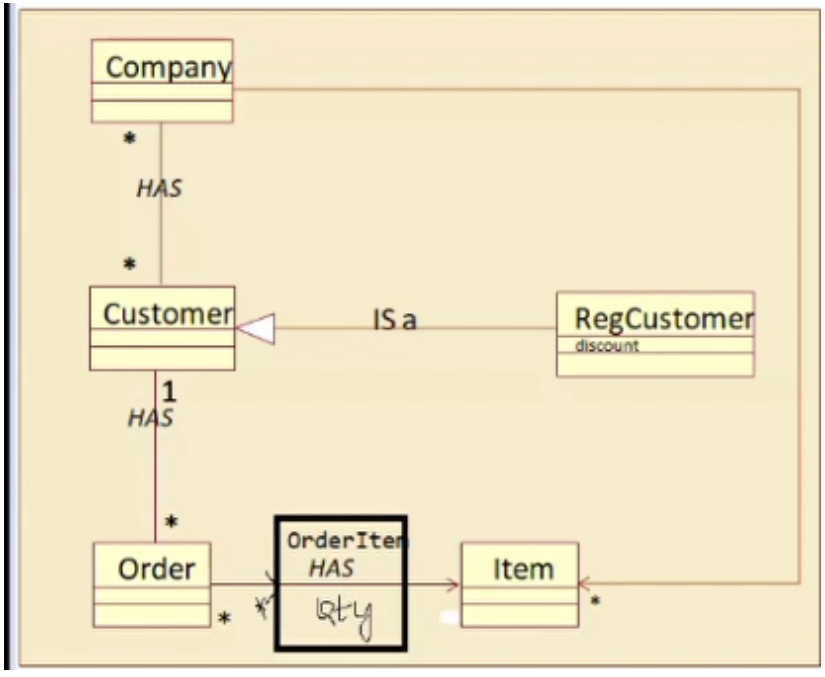


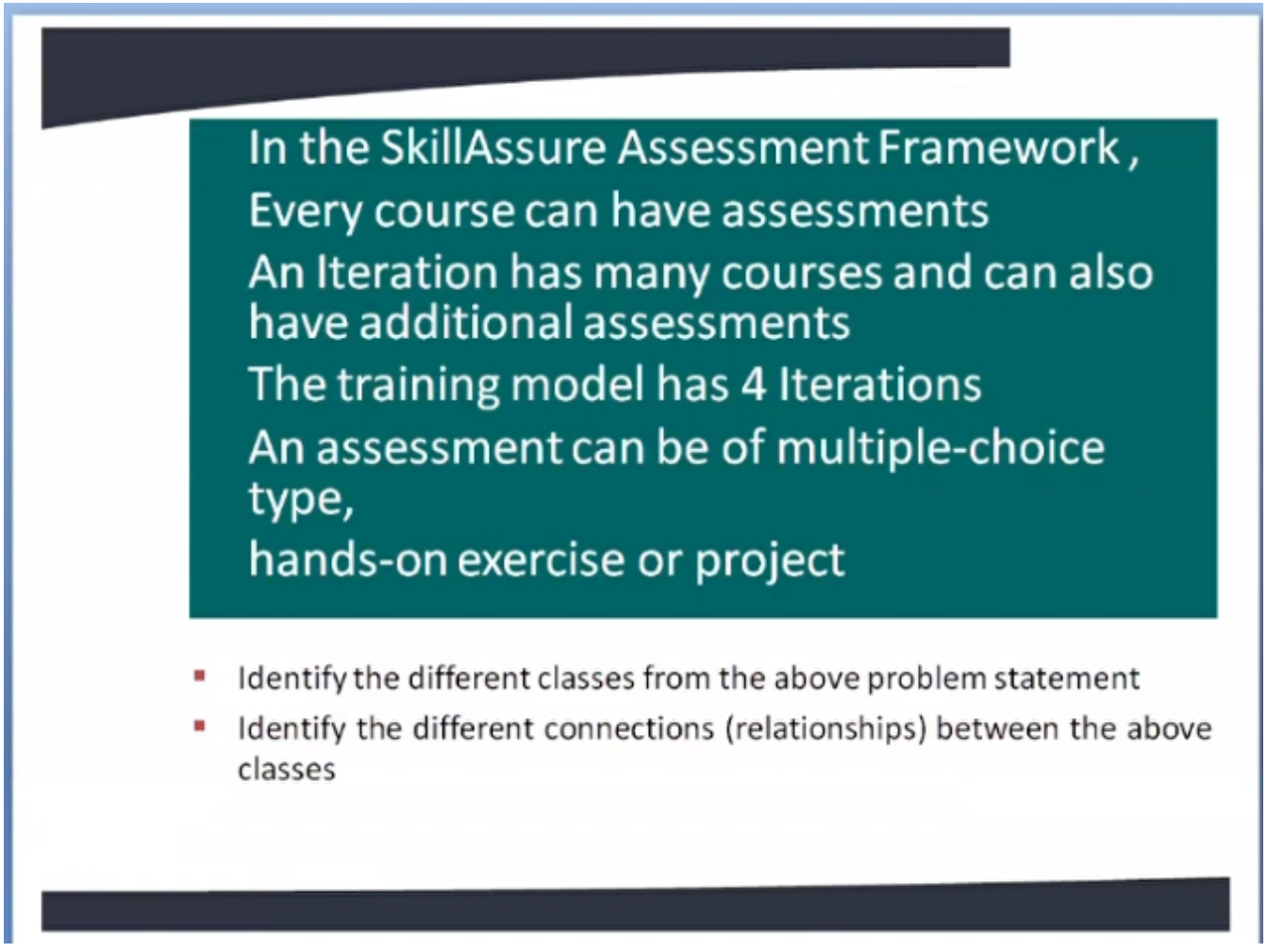
The OO Model



A Customer can place many orders implies that RegCustomer can also place many Orders.

A Company has many Customers implies that a Company also has many RegCustomers


Company
*
HAS
*
Customer
IS a
RegCustomer
discount
1
HAS
*
Order
OrderIter
HAS
Qty
Item
*
*



Assignment - Write a program for the above ques

Exercise

There are many programming languages. Java and C# are object-oriented programming languages. C is a procedural programming language.

- Identify the different classes from the above problem statement
- Identify the different connections (relationships) between the above classes

Let us program Object using C# as language

What Is a Class?

- For the philosopher…
  - An artefact of human *class*ification!
  - *Class*ify based on common behaviour or attributes
  - Agree on descriptions and names of useful *class*es
  - Create vocabulary; we communicate; we think!

- For the object-oriented programmer…
  - A named syntactic construct that describes common behaviour and attributes
  - A data structure that includes both data and functions

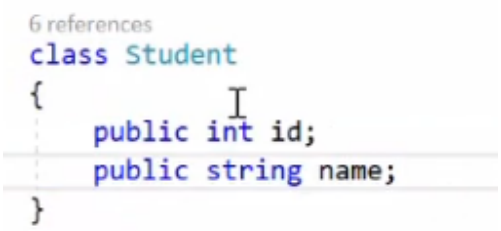
Defining a Class

- A **class** acts as the template from which an instance of an *object* is created. The class defines the **properties** of the object and the **methods** used to control the object's behavior.
- A **class** specifies the **structure** of **data** as well as the **methods** which manipulate that **data**. Such data and methods are contained in each instance of the class.
- A **class** is a **model** or **template** that can be instantiated to create **objects** with a common definition, and therefore common properties, operations and behavior.
- A **class** provides a **template** for defining the behavior of a particular type of object. **Objects** are referred to as "**instances**" of a class.

Everything is an Object

What is an Object?

Anything that you can describe can be represented as an object, and that representation can be created, manipulated and destroyed to represent how you use the real object that it models.

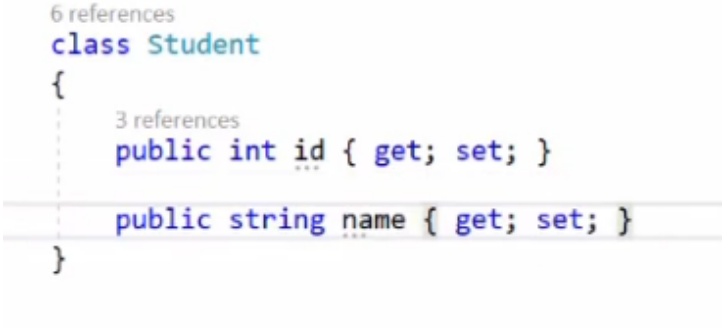


IS this a field or a property ?

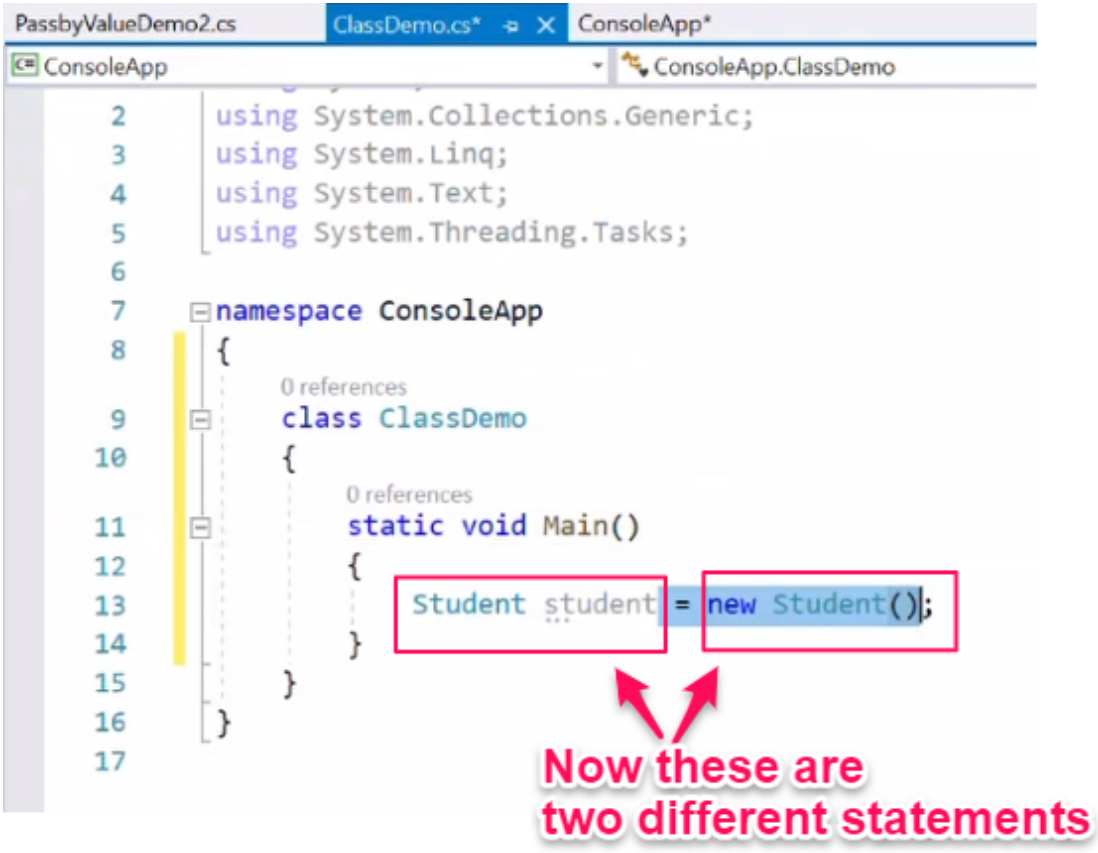
--it is a field.

How will i change this to property ?

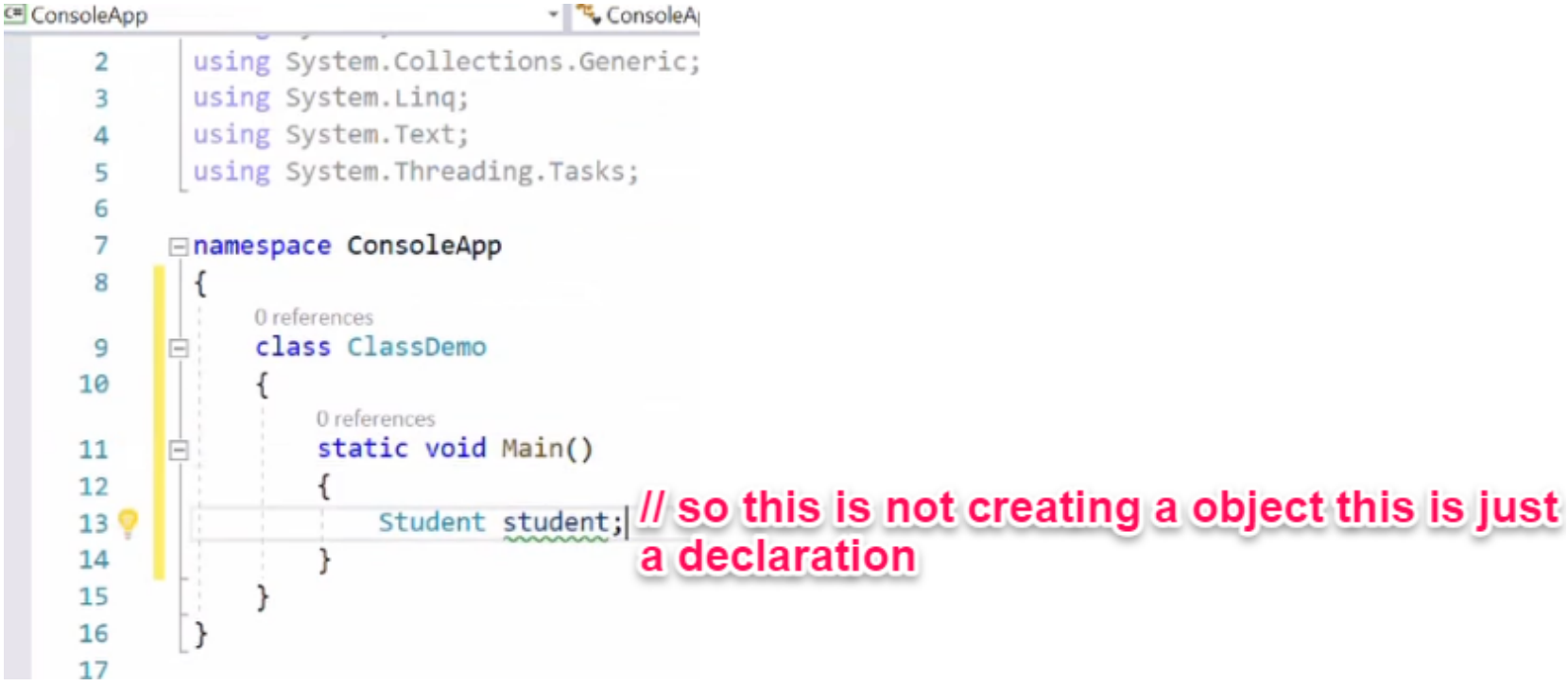
--you need to put just a get and set method

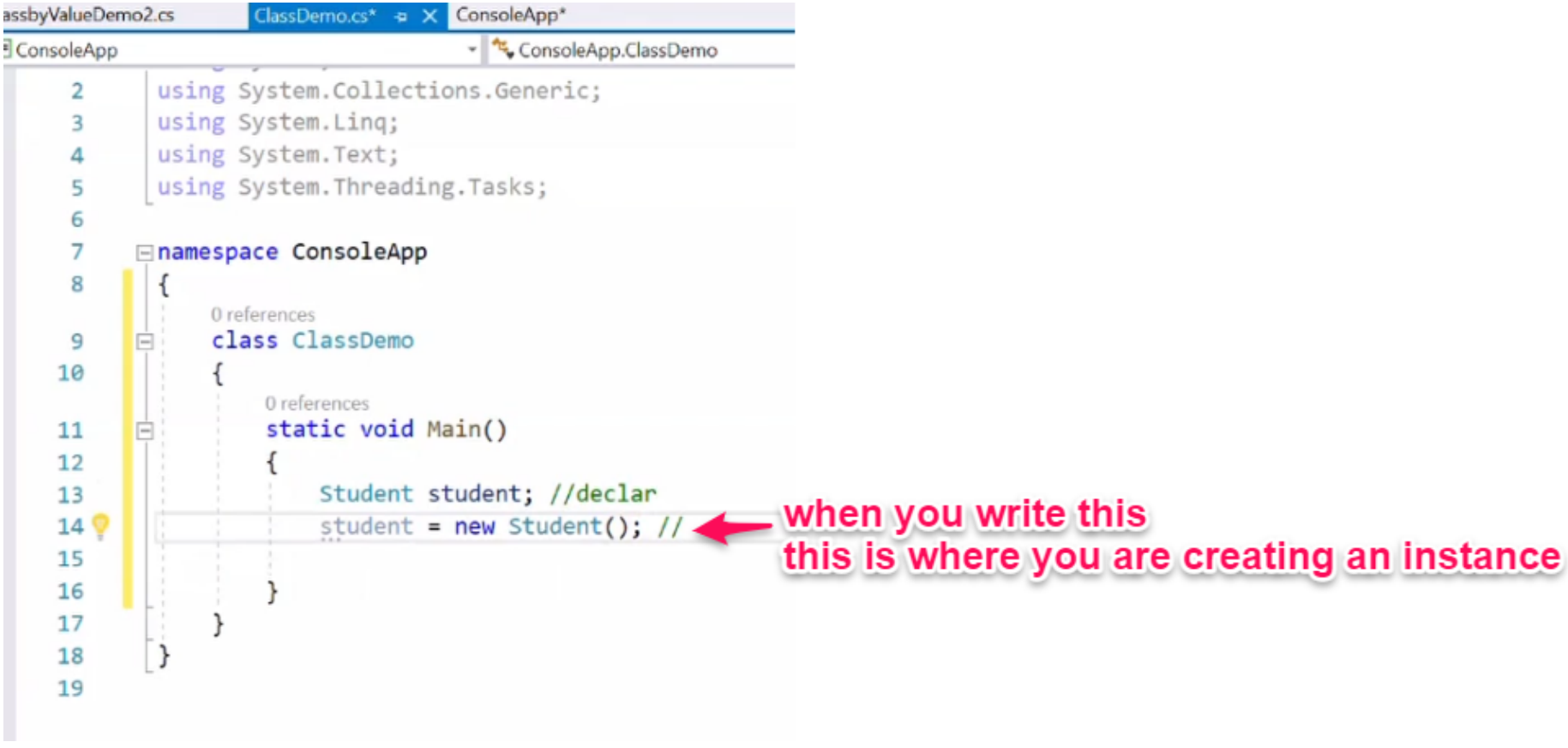


now we need to rename this because property should always be in capitals



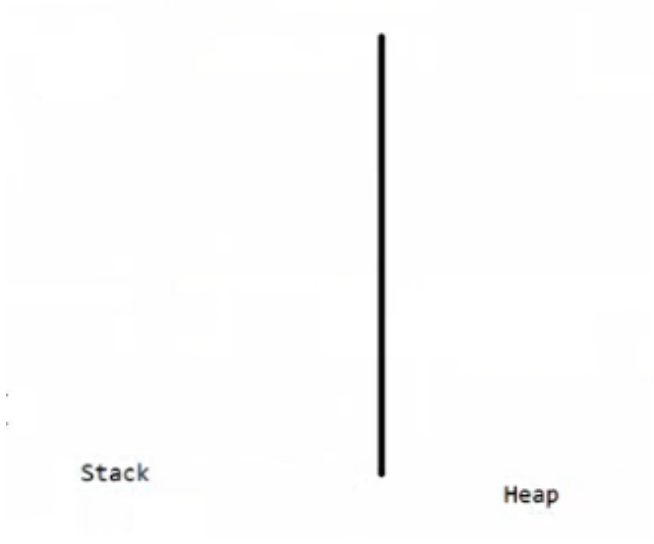
So i can just specify like this


ConsoleApp
ConsoleA
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApp
{
0 references
class ClassDemo
{
0 references
static void Main()
{
Student student;
}
}
}
// so this is not creating a object this is just a declaration



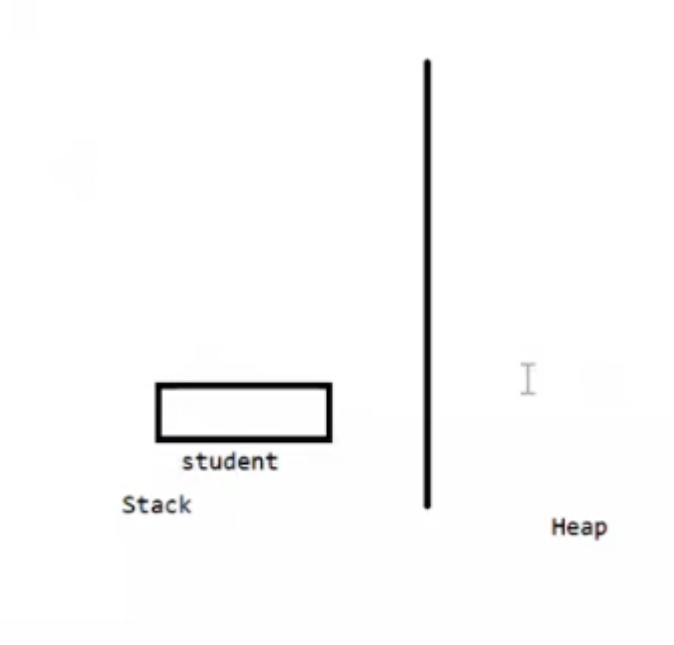
Now whenever you creating a particular instance, **So there are 6 different steps which runs internally**

So internally the computers memory will have Stack and Heap



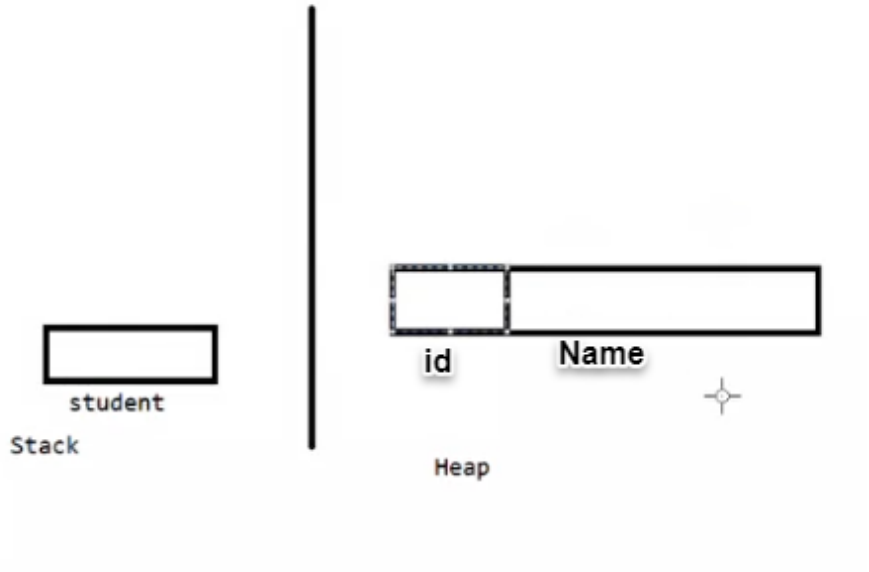
So

1. in the stack it gone get address created, i.e the declaration Student student; gets executed



2. = new Student() will get me the instance of this student created, now student will have the data members ---Id and Name

so we'll have the id and name defined in the Heap



3. this id and name gets initialized --**Instance Initialization**

So id will be initialized to 0

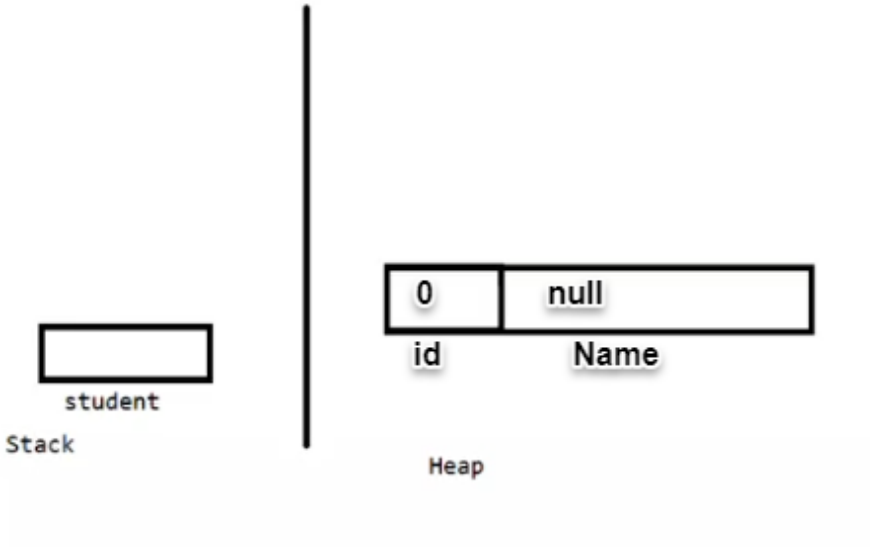
and the string will be initialize to empty or null, because its a reference type

remember all value type stored with the default value and all the reference type will be stored to null

for ex;- if i take a numeric value all of them will be initialize to 0

boolean will be initialize to false

and all the reference type will be initialize to null

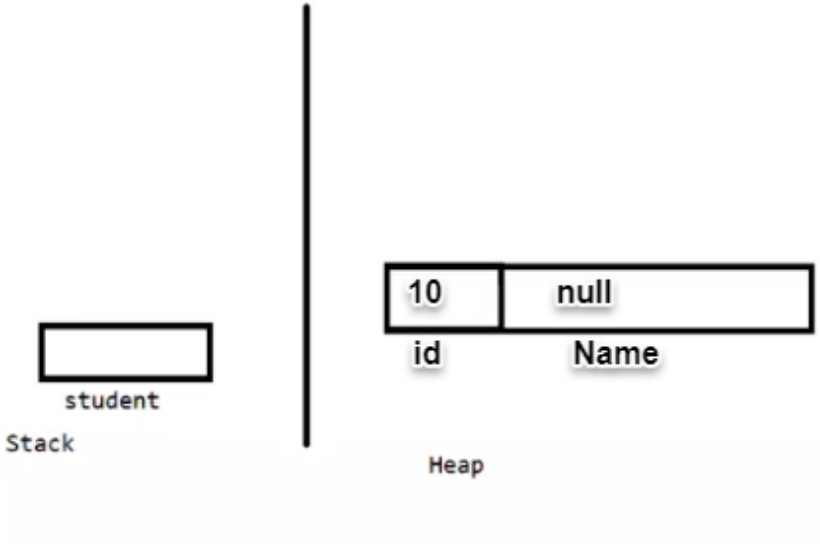


4. if you have explicit initialization that's get executed

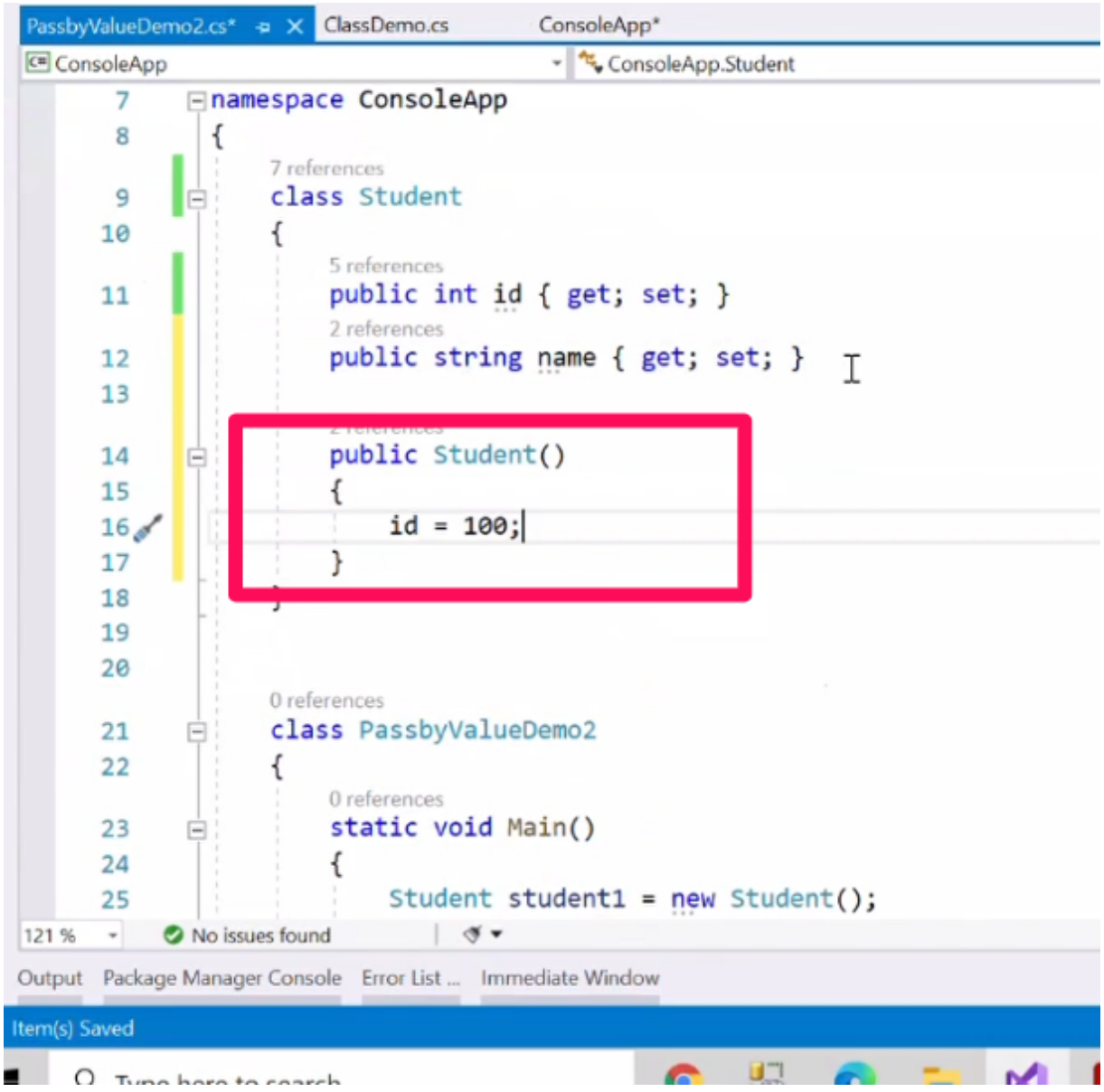
```
6 references
class Student
{
    3 references
    public int id { get; set; } = 10;
    2 references
    public string name { get; set; }
}
```

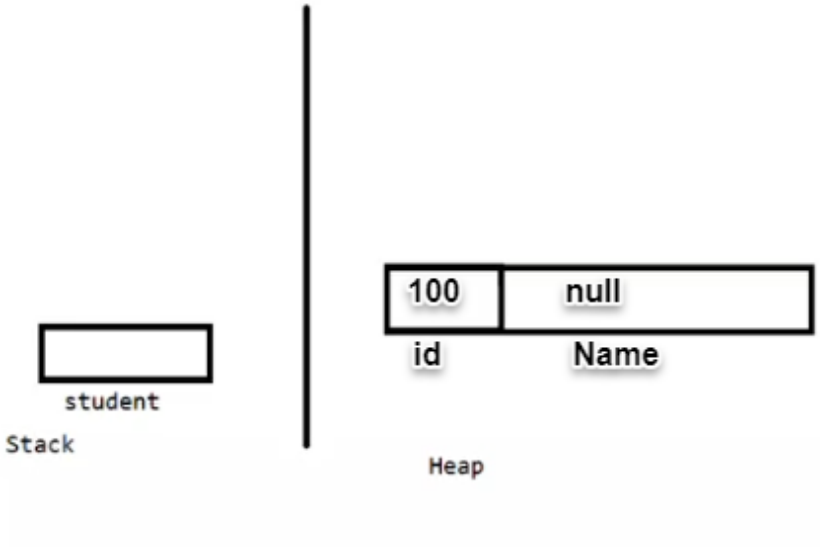
So whenever you look into a particular class, So your class may have initialization something like this i'm initializing my id with 10

this is called Explicit initialization

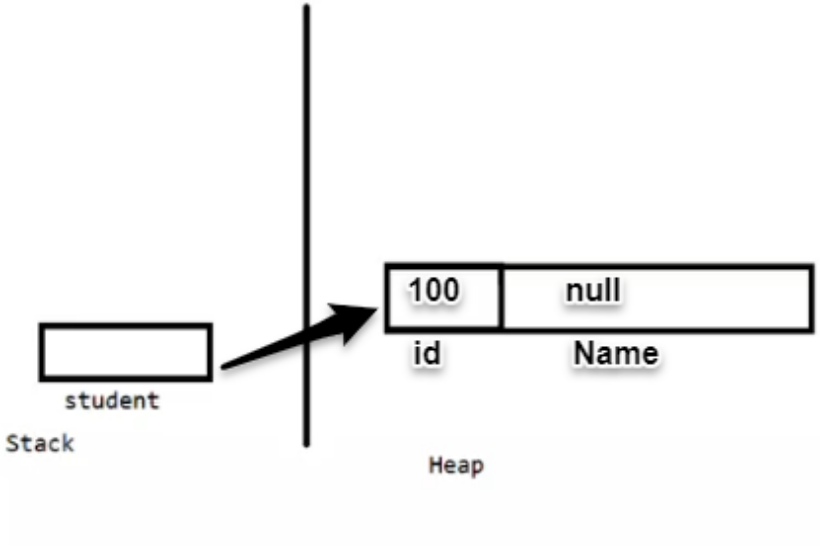


5. if you have a constructor that's get executed

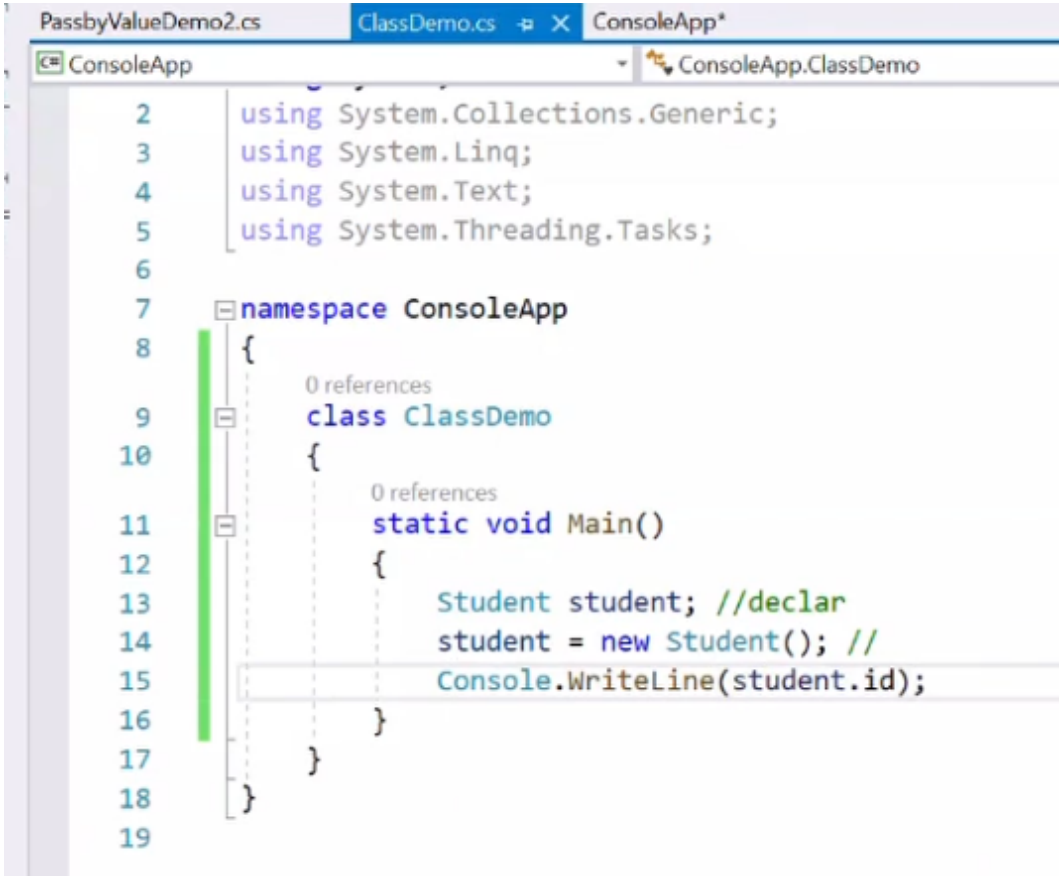

PassbyValueDemo2.cs*
ClassDemo.cs
ConsoleApp*
ConsoleApp
ConsoleApp.Student
namespace ConsoleApp
{
7 references
class Student
{
5 references
public int id { get; set; }
2 references
public string name { get; set; }
public Student()
{
id = 100;
}
}
0 references
class PassbyValueDemo2
{
0 references
static void Main()
{
Student student1 = new Student();
121 %
No issues found
Output
Package Manager Console
Error List ...
Immediate Window
Item(s) Saved



6. the base address of this will be returned back to student



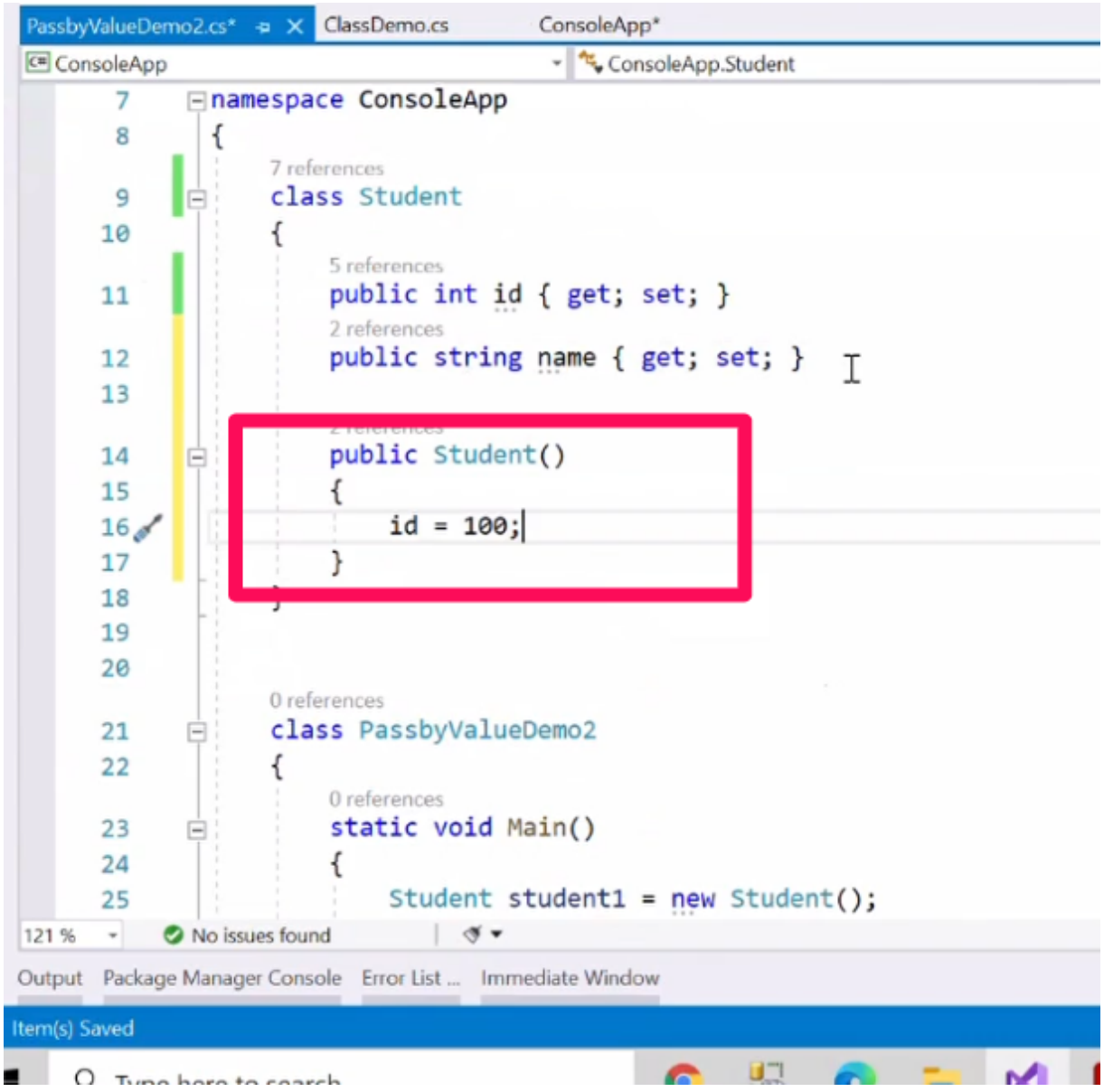
You can see this when i try to print this



so it print me 0 here

In java and .Net there in no concept of garbage value

So Who have initialized that somebody has initialized it right otherwise how will it get value 0
So you may have constructor , may be you would have initialize it with some value to this variable

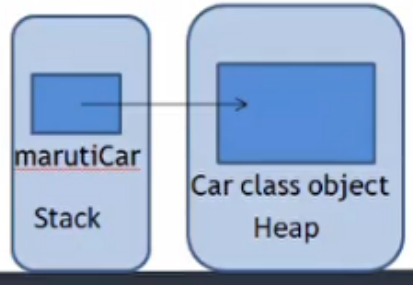

PassbyValueDemo2.cs*
ClassDemo.cs
ConsoleApp*
ConsoleApp
ConsoleApp.Student
namespace ConsoleApp
{
7 references
class Student
{
5 references
public int id { get; set; }
2 references
public string name { get; set; }
public Student()
{
id = 100;
}
}
0 references
class PassbyValueDemo2
{
0 references
static void Main()
{
Student student1 = new Student();
121 %
No issues found
Output
Package Manager Console
Error List ...
Immediate Window
Item(s) Saved

How to Create an Instance of a Class in C#?

- Declaring a class variable does not create an object
  - Use the **new** operator to create an object
  - In memory, heap will contain the object and stack will have the reference of the object

```
class Program
{
    static void Main()
    {
        Car marutiCar = new Car();
        marutiCar.StartCar();
        marutiCar.stopCar();

    }
}
```

```
Car started..
car stopped..
```



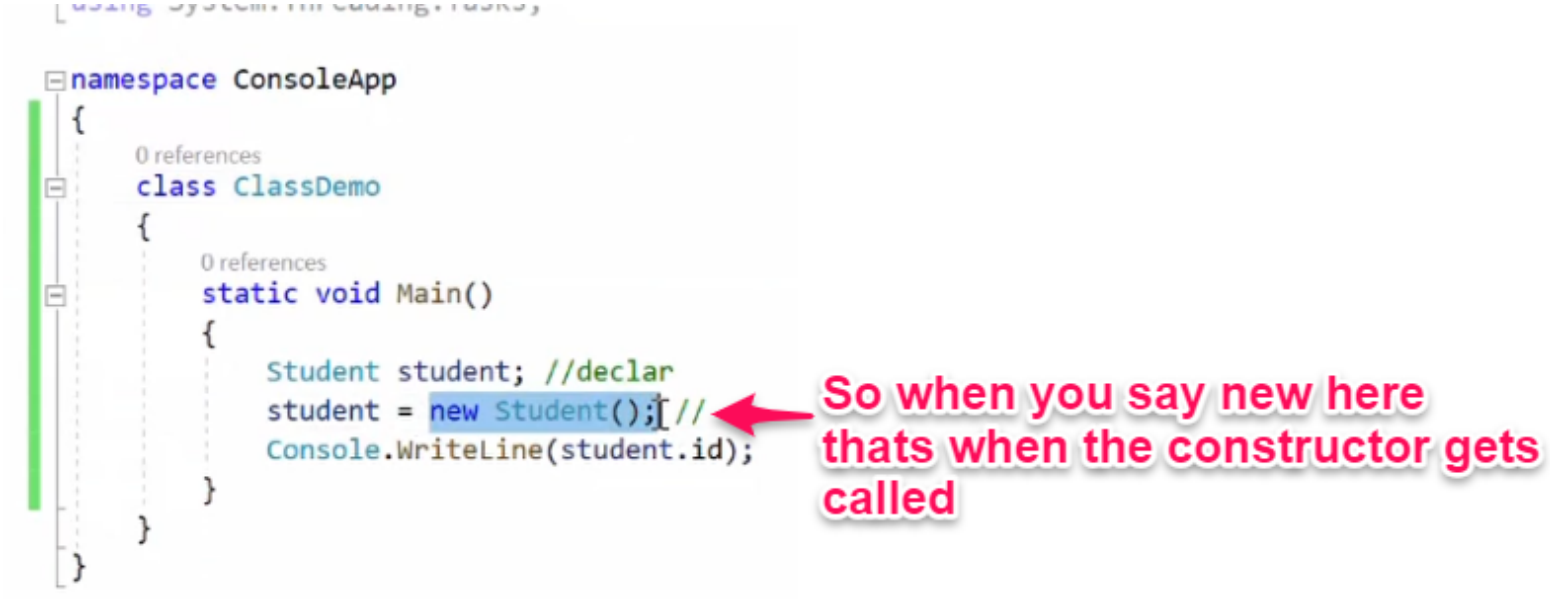
Stack: Section of process memory space where memory spaces for local variables (local to the method, such as Main method) are allocated

Constructor

- A constructor is a special method that is called to initialize default values to data members (fields) of a class
- It does not create object
- Constructor's name is same as that of the class.
- Constructor does not return anything

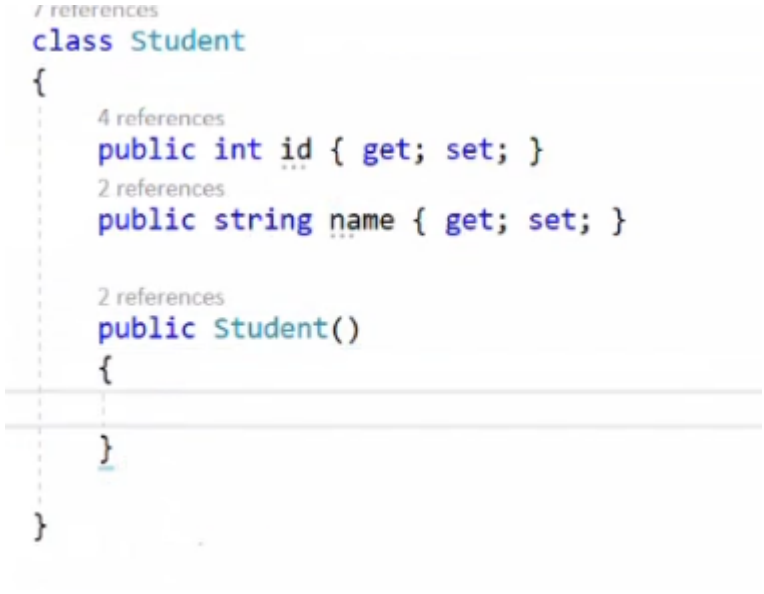
```
class Car
{
    string modelName;
    bool hasAc;
    string bodyColor;

    public Car()
    {
        modelName = "Maruti";
        hasAc = true;
        bodyColor = "Blue";
    }

    public void StartCar()
    {
        Console.WriteLine("Car started..");
    }
    public void stopCar()
    {
        Console.WriteLine("car stopped..");
    }
}
```

Constructor

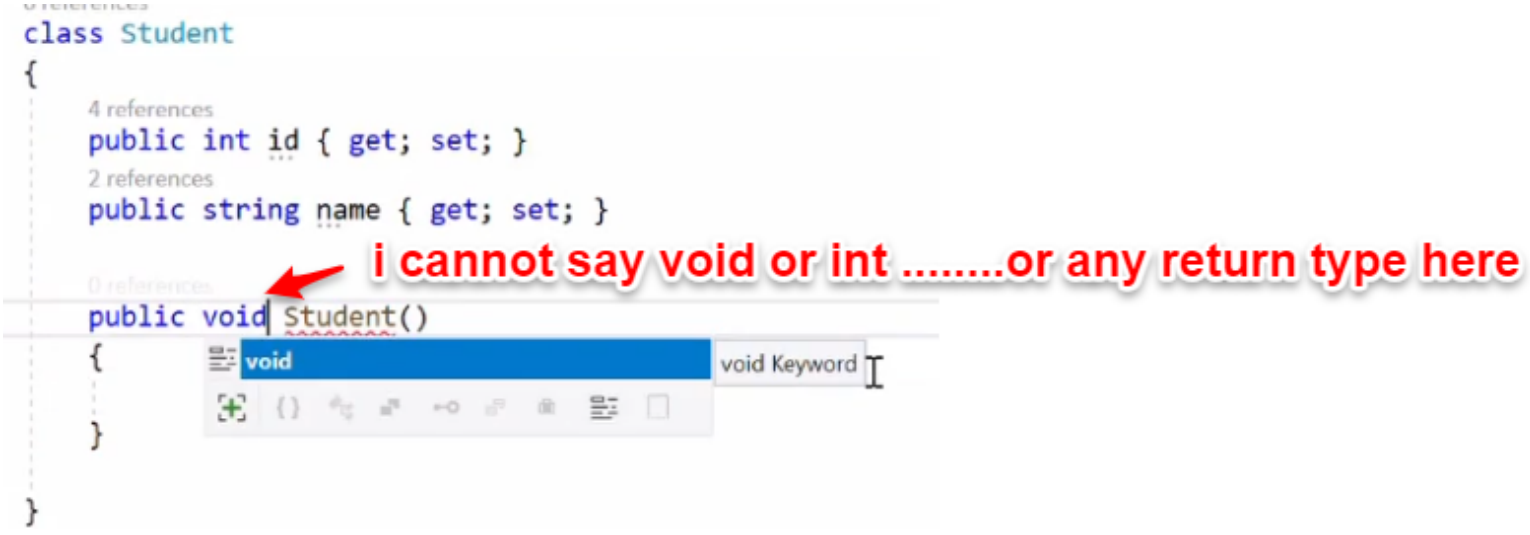


Constructor are Special methods which will automatically get called when a new object is created

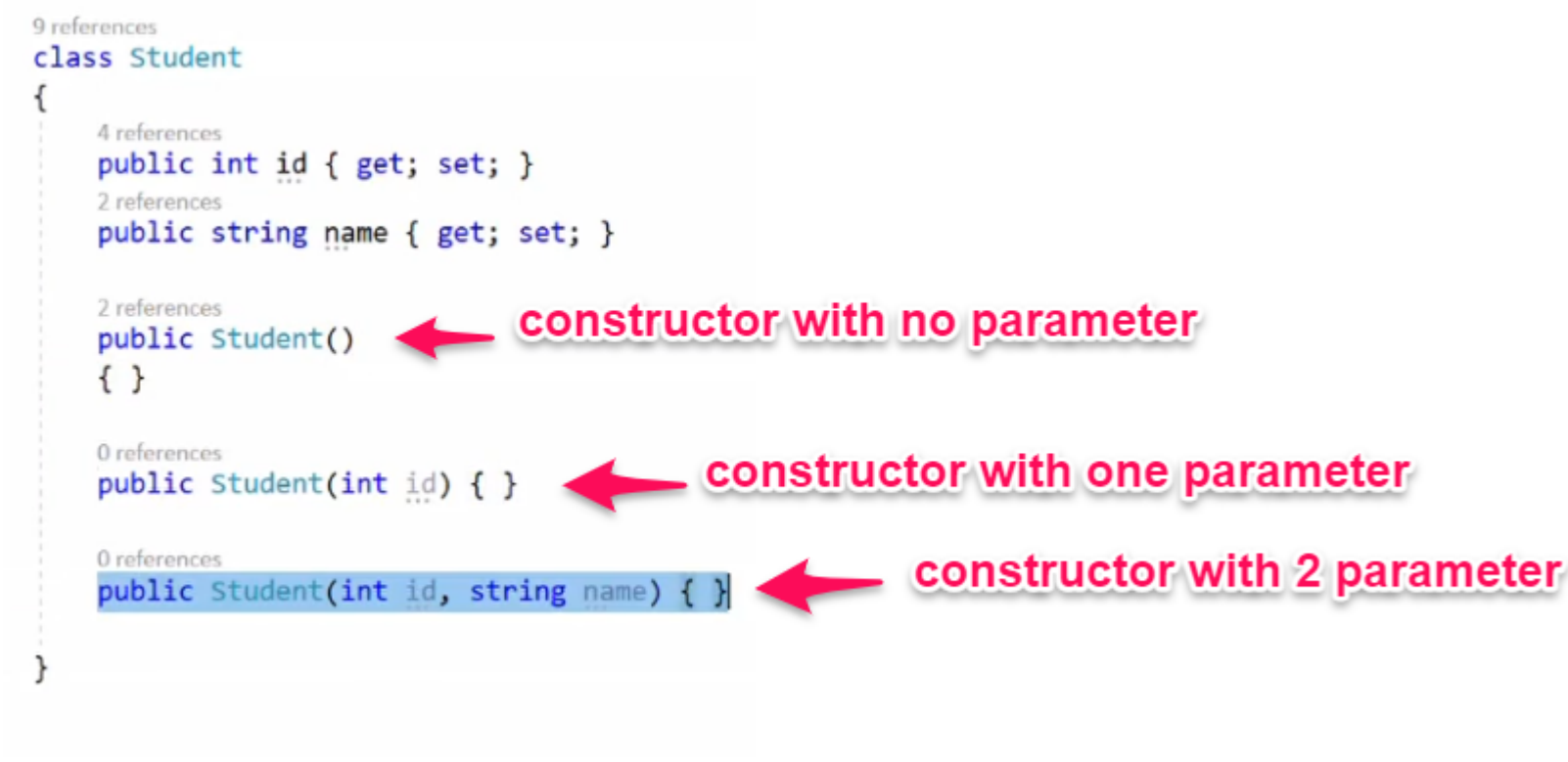
1. constructor should have same name as class name



2. constructor should not have any return value


class Student
{
4 references
public int id { get; set; }
2 references
public string name { get; set; }
i cannot say void or int .....or any return type here
0 references
public void Student()
{
void
void Keyword
}
}

3. we can have multiple constructors in a class but atleast one parameter should change



4. Constructors are special methods but we cannot call them, we cannot explicitly call a constructor

   we cannot say student.student not possible

   they are called only once when the object is created

5. We have something called **Default Constructor**

   Default constructors are those which are provided by the system, whenever you create a class by default a constructor will be given to you

   Now whenever you write your own constructor

   Shortcut to create constructor

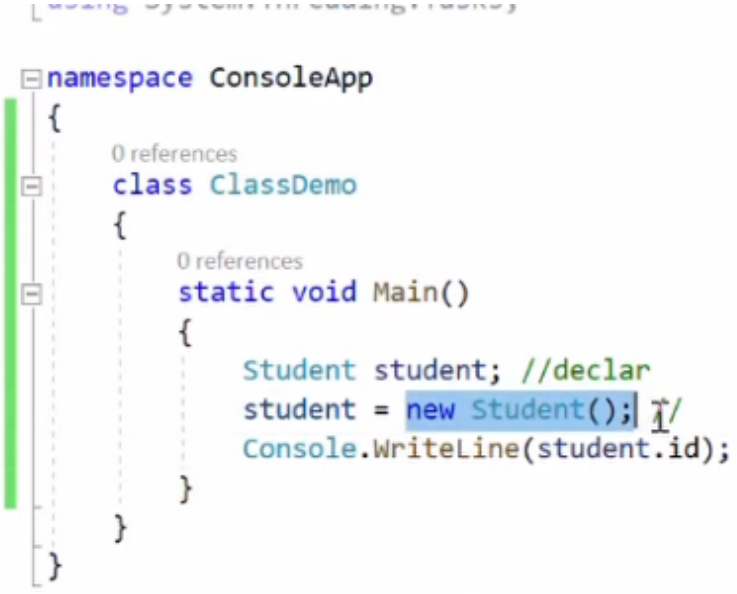
write **ctor then press tab tab** it will automatically create a constructor for you

when you write your own constructor the default constructor gets overloaded that will be removed and this constructor gets executed

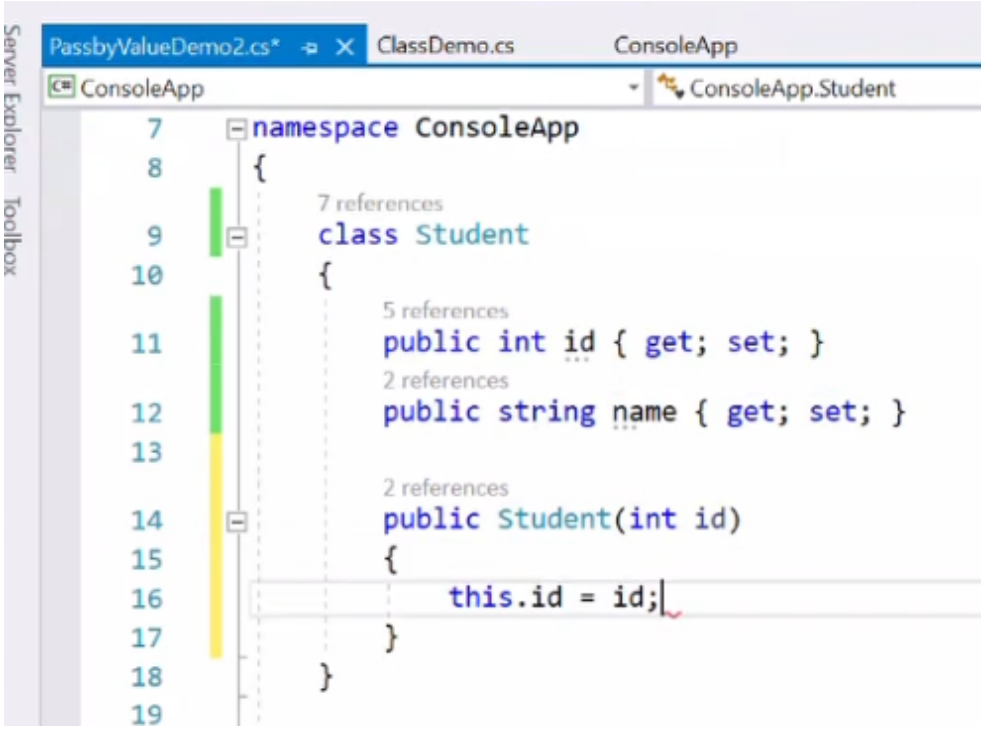
What does it mean?

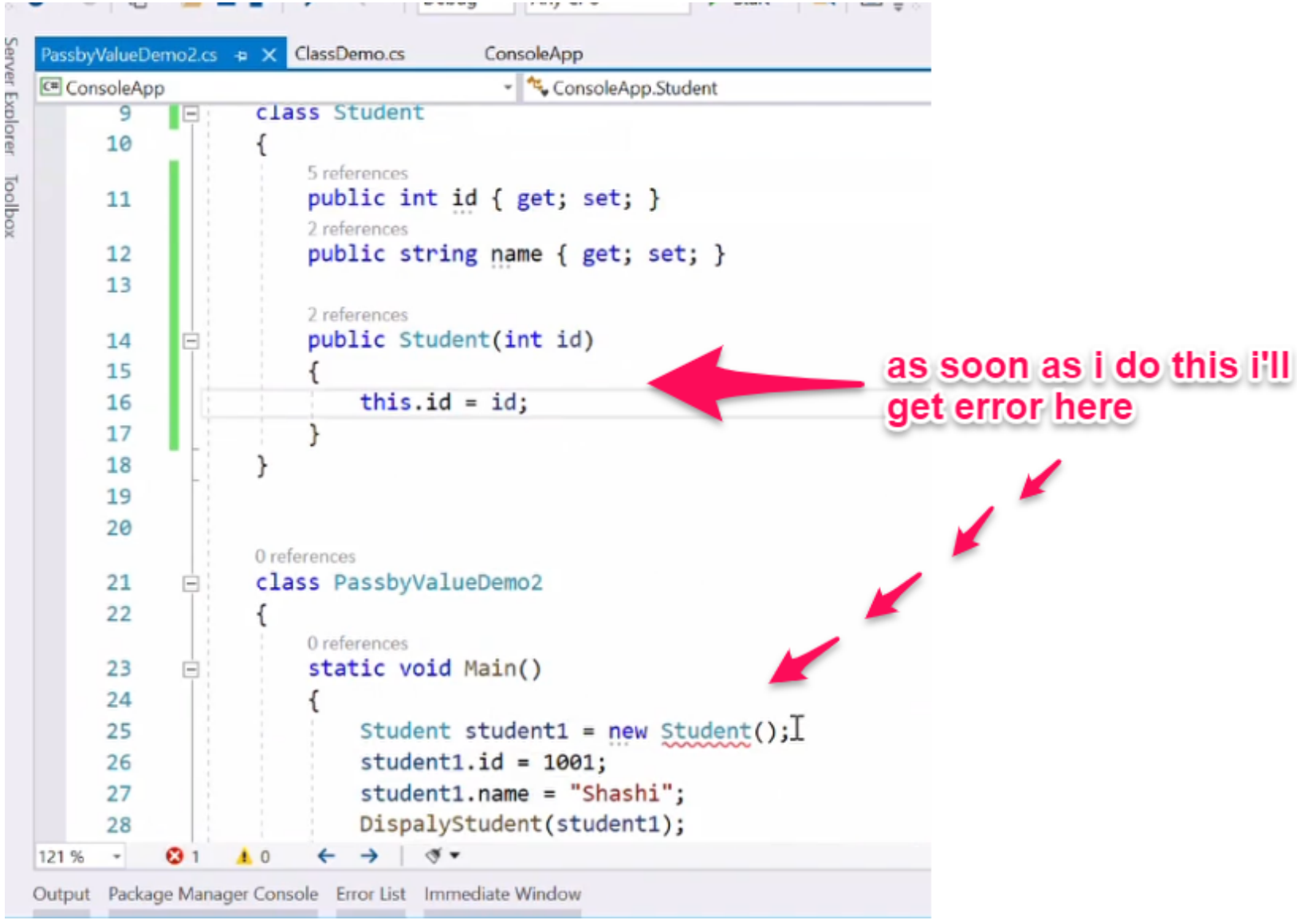
Can you see i'm saying = new Student();

here

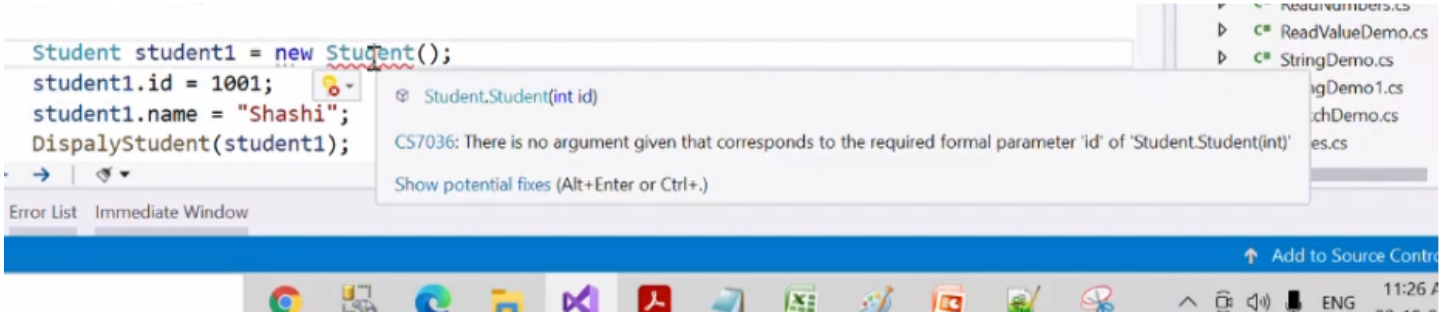


as soon as i make this as int id and i'll say this.id = id;


Server Explorer
Toolbox
PassbyValueDemo2.cs*
ClassDemo.cs
ConsoleApp
ConsoleApp
ConsoleApp.Student
7
8
9
10
11
12
13
14
15
16
17
18
19
namespace ConsoleApp
{
7 references
class Student
{
5 references
public int id { get; set; }
2 references
public string name { get; set; }
2 references
public Student(int id)
{
this.id = id;
}
}


PassbyValueDemo2.cs
ClassDemo.cs
ConsoleApp
ConsoleApp
ConsoleApp.Student
class Student
{
5 references
public int id { get; set; }
2 references
public string name { get; set; }
2 references
public Student(int id)
{
this.id = id;
}
}
0 references
class PassbyValueDemo2
{
0 references
static void Main()
{
Student student1 = new Student();
student1.id = 1001;
student1.name = "Shashi";
DispalyStudent(student1);
as soon as i do this i'll get error here
121 %
1
0
Output
Package Manager Console
Error List
Immediate Window
Server Explorer
Toolbox

it will say boss you don't have a constructor which will take one argument

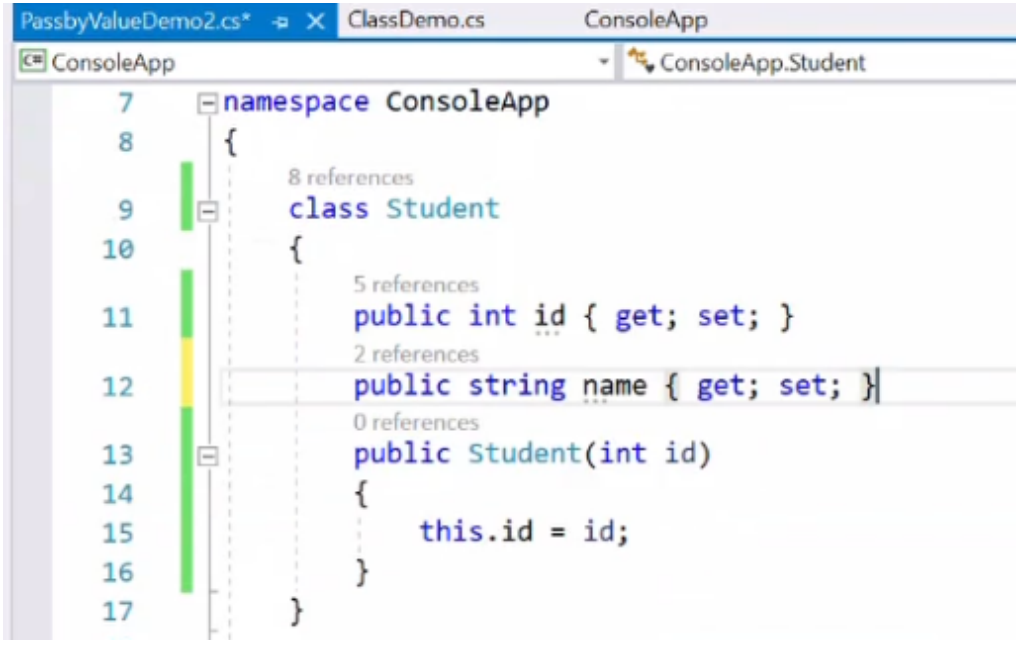


meaning the default constructor which is provided by the system is not there

SO, if you are creating constructor then its your responsibility as a developer to give a constructor for the user

So Why should we use a constructor ?

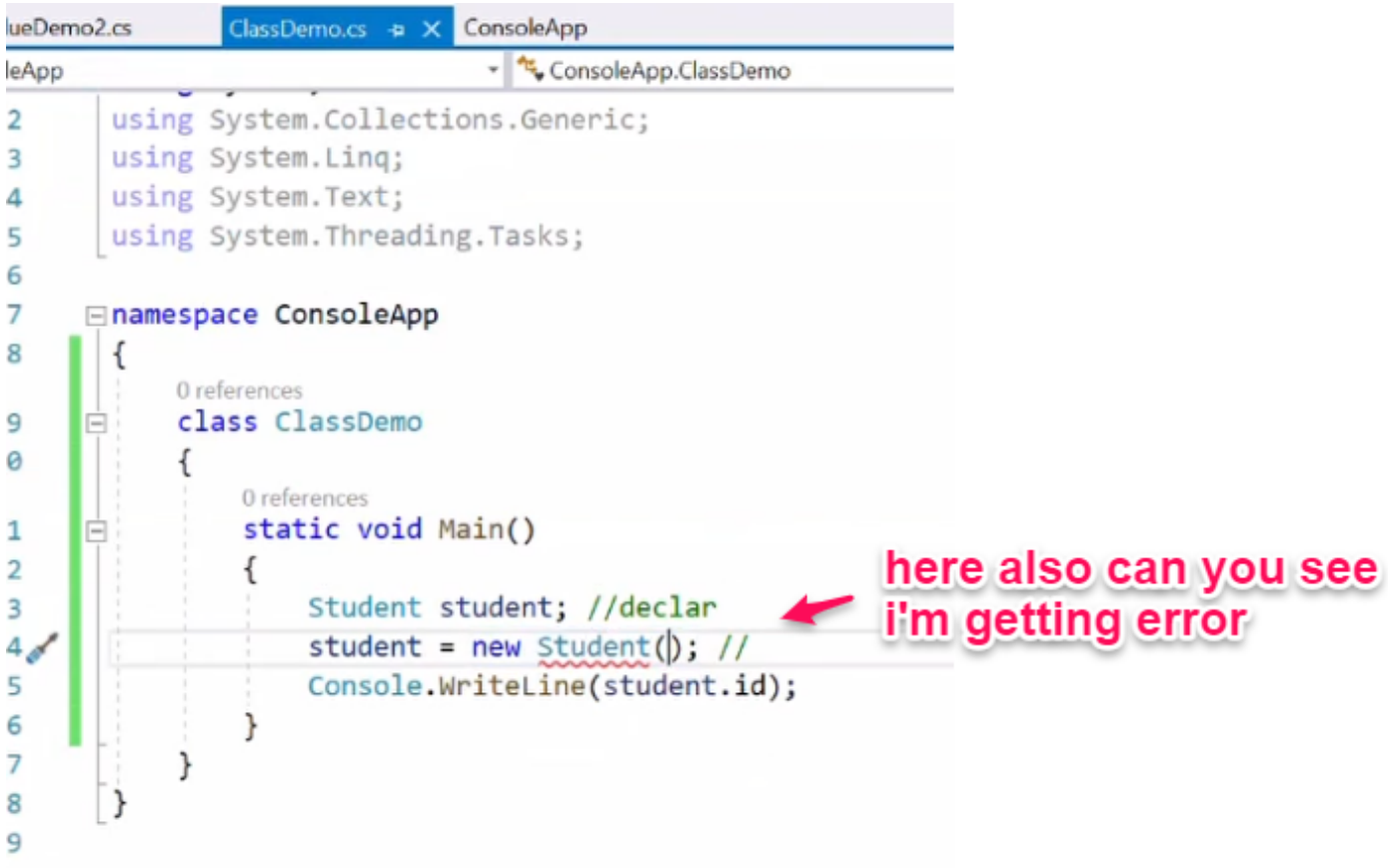
so constructor are use to initialize your data members, as soon as i create a object i want to put some value to that object for Example id may be like as soon as i create a employee object immediately i want to put an id i don't want user to enter the id i want to create that particular id
now imagine if you don't pass the id i don't want anybody to create instance of my variable

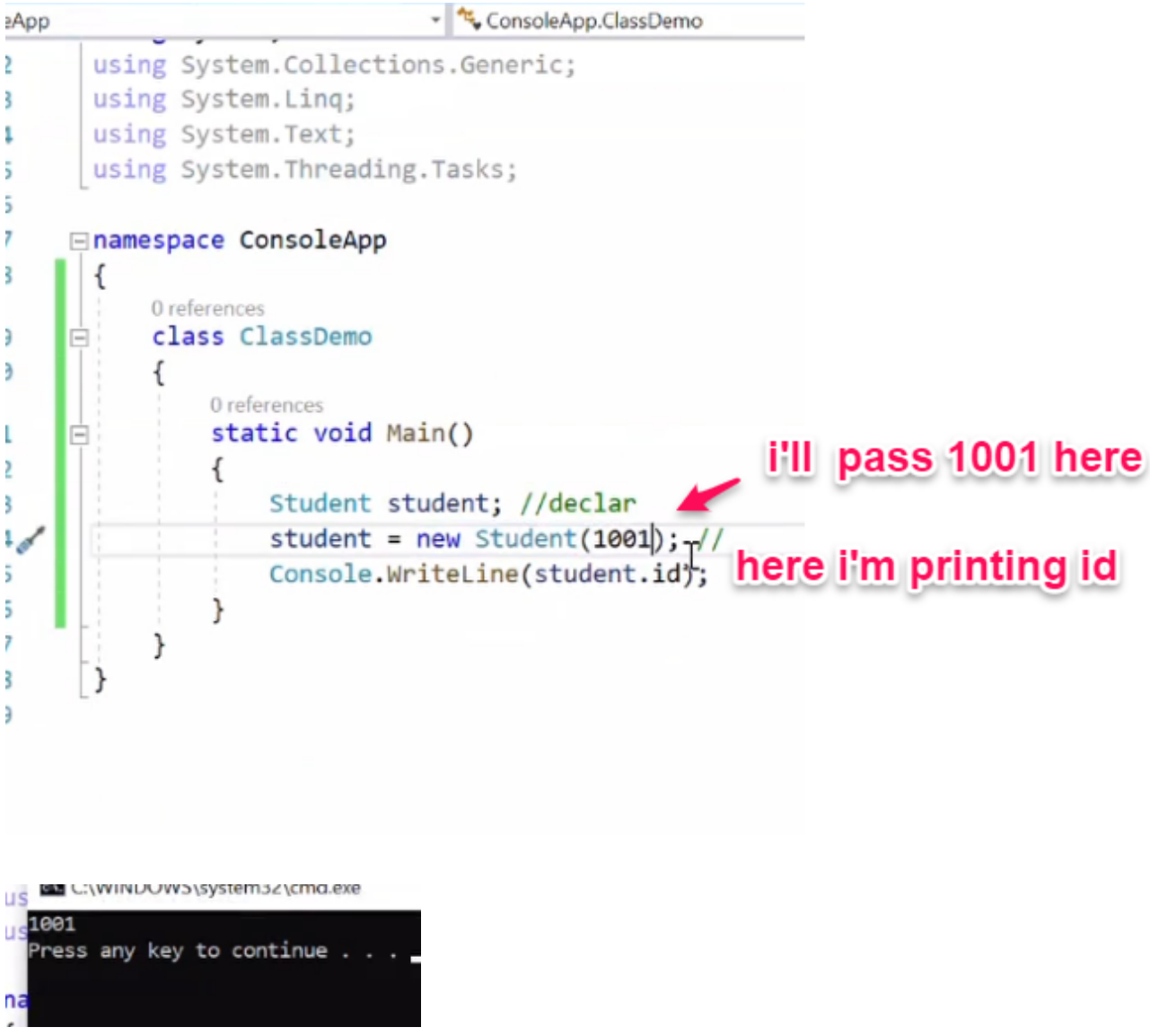


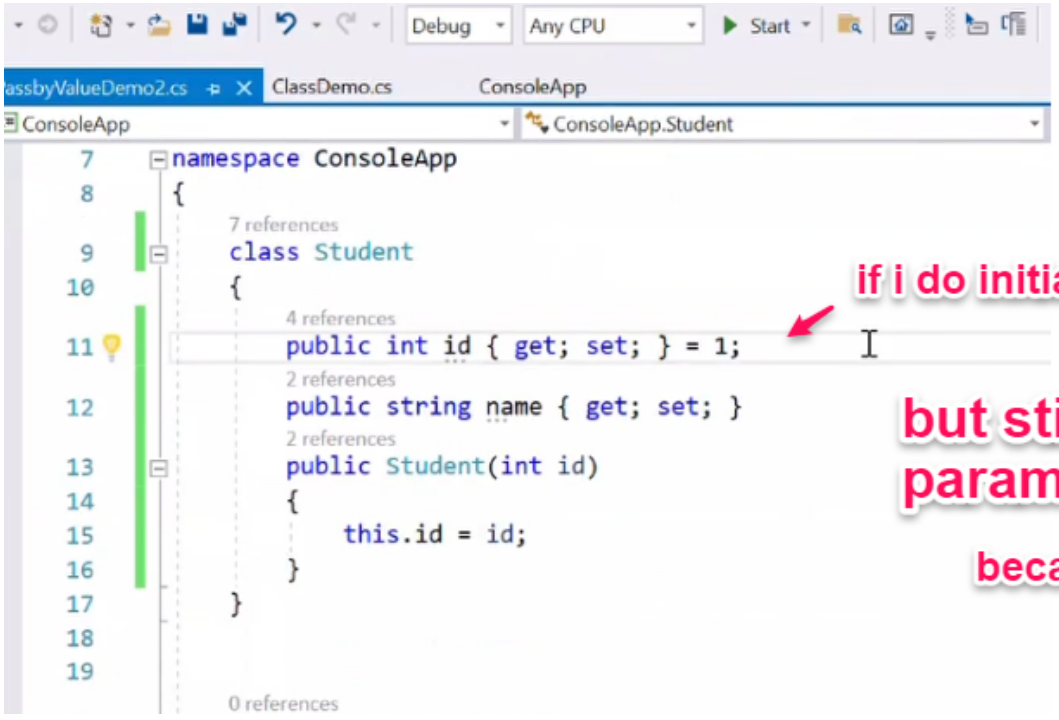
```
}


0 references
class PassbyValueDemo2
{
    0 references
    static void Main()
    {
        Student student1 = new Student();
        student1.id = 1001;
        student1.name = "Shashi";
        DispalyStudent(student1);
        Student student2 = student1;
        student2.id = 1002;
        DispalyStudent(student2);
        DispalyStudent(student1);
    }
```

you can see i'm already getting an error here


0 references
class PassbyValueDemo2
{
0 references
static void Main()
{
Student student1 = new Student(1001);
//student1.id = 1001;
student1.name = "Shashi";
DispalyStudent(student1);
Student student2 = student1;
student2.id = 1002;
DispalyStudent(student2);
DispalyStudent(student1);
}
so instead of me saying student.id here i'll pass 1001 here
as a parameter


lueDemo2.cs
ClassDemo.cs
ConsoleApp
leApp
ConsoleApp.ClassDemo
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApp
{
0 references
class ClassDemo
{
0 references
static void Main()
{
Student student; //declar
student = new Student(); //
Console.WriteLine(student.id);
}
}
}
here also can you see
i'm getting error

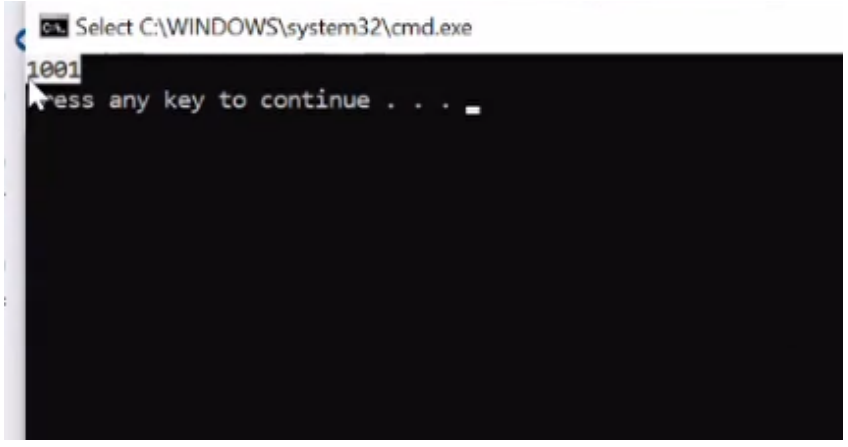

eApp
ConsoleApp.ClassDemo
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApp
{
0 references
class ClassDemo
{
0 references
static void Main()
{
Student student; //declar
student = new Student(1001);//
Console.WriteLine(student.id);
}
}
}
i'll pass 1001 here
here i'm printing id
C:\WINDOWS\system32\cmd.exe
1001
Press any key to continue . . .



if i do initialization here

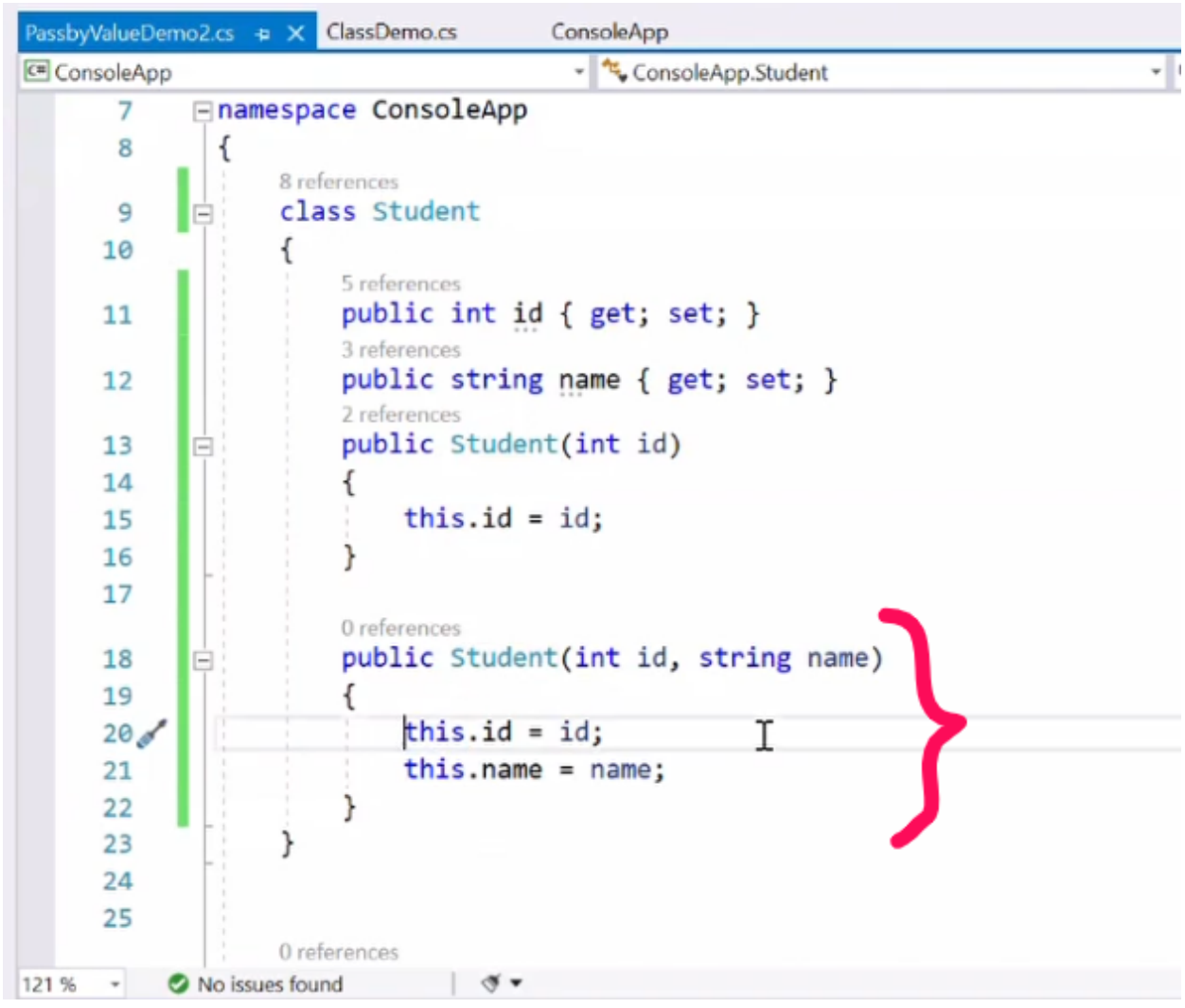
but still i'll get what i'm passing as parameter

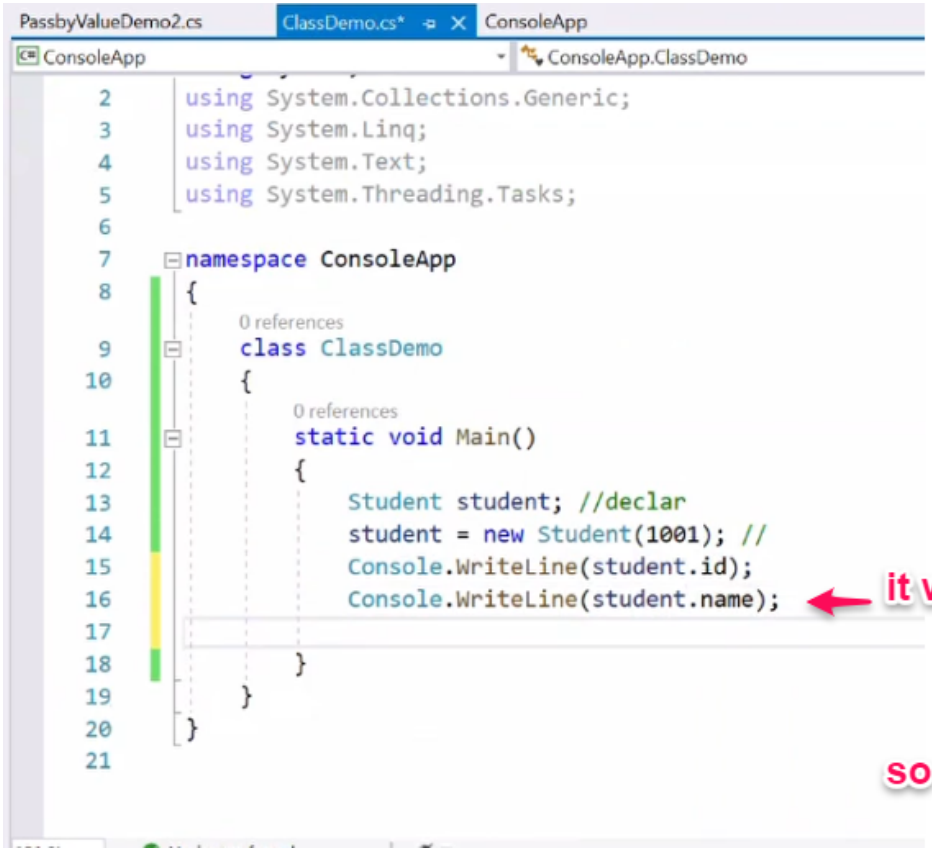
because it is written inside the constructor



Now suppose i want to give more flexibility to the user that user can pass the particular id and then i'll initialize the particular id

I want to give preference to the user where based on the preference he can either pass id or he can pass name also i'm giving him the flexibility to initialize both my id and name

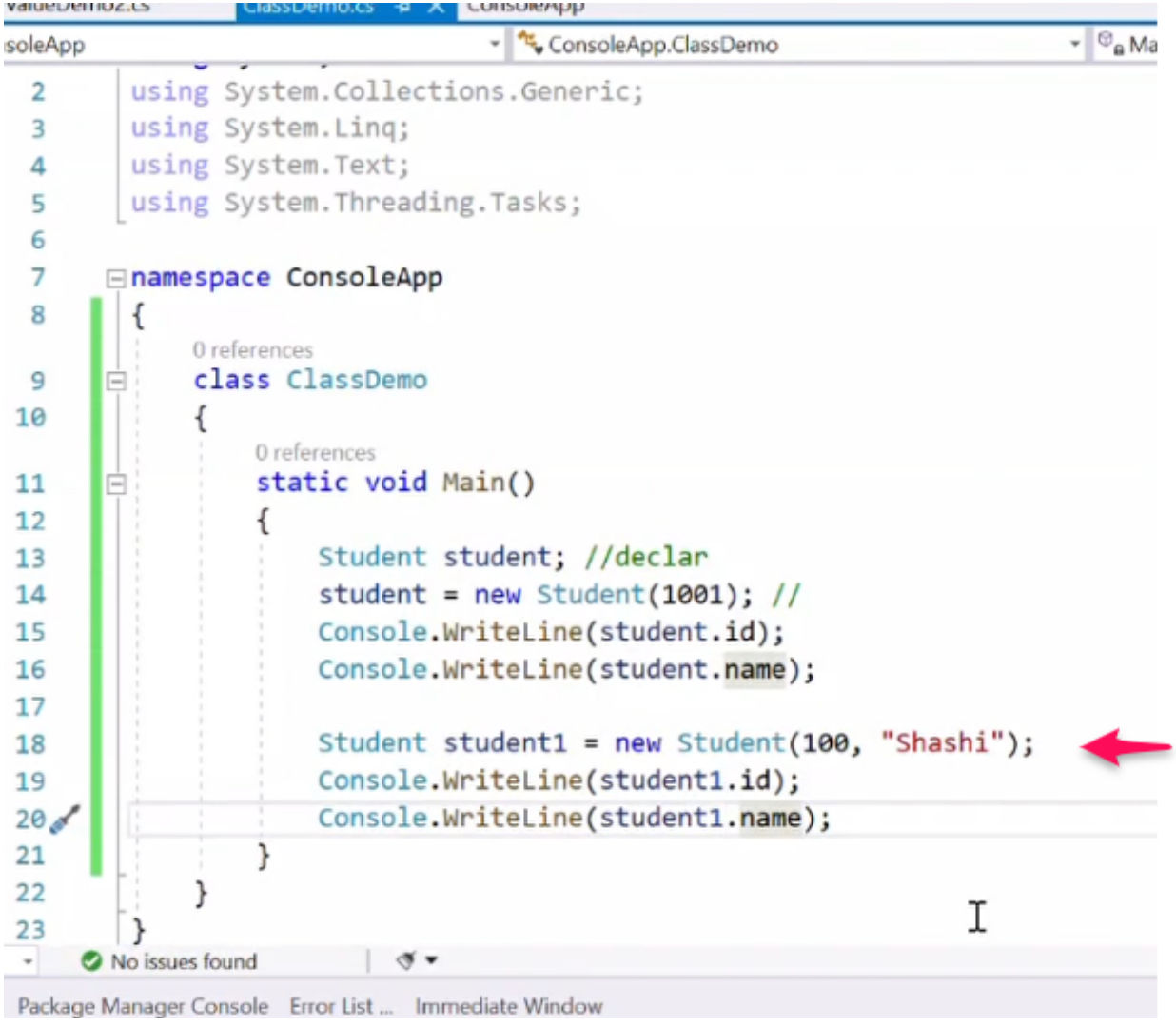

PassbyValueDemo2.cs
ClassDemo.cs
ConsoleApp
ConsoleApp
ConsoleApp.Student
namespace ConsoleApp
{
8 references
class Student
{
5 references
public int id { get; set; }
3 references
public string name { get; set; }
2 references
public Student(int id)
{
this.id = id;
}
0 references
public Student(int id, string name)
{
this.id = id;
this.name = name;
}
}
0 references
121 %
No issues found



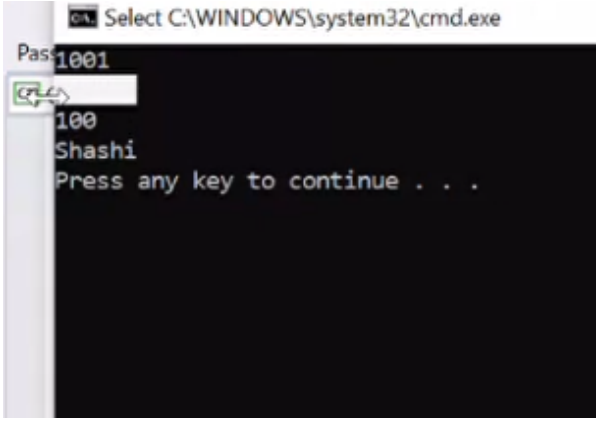
it will print me nothing

because i'm not initializing the particular name

so lets create a student



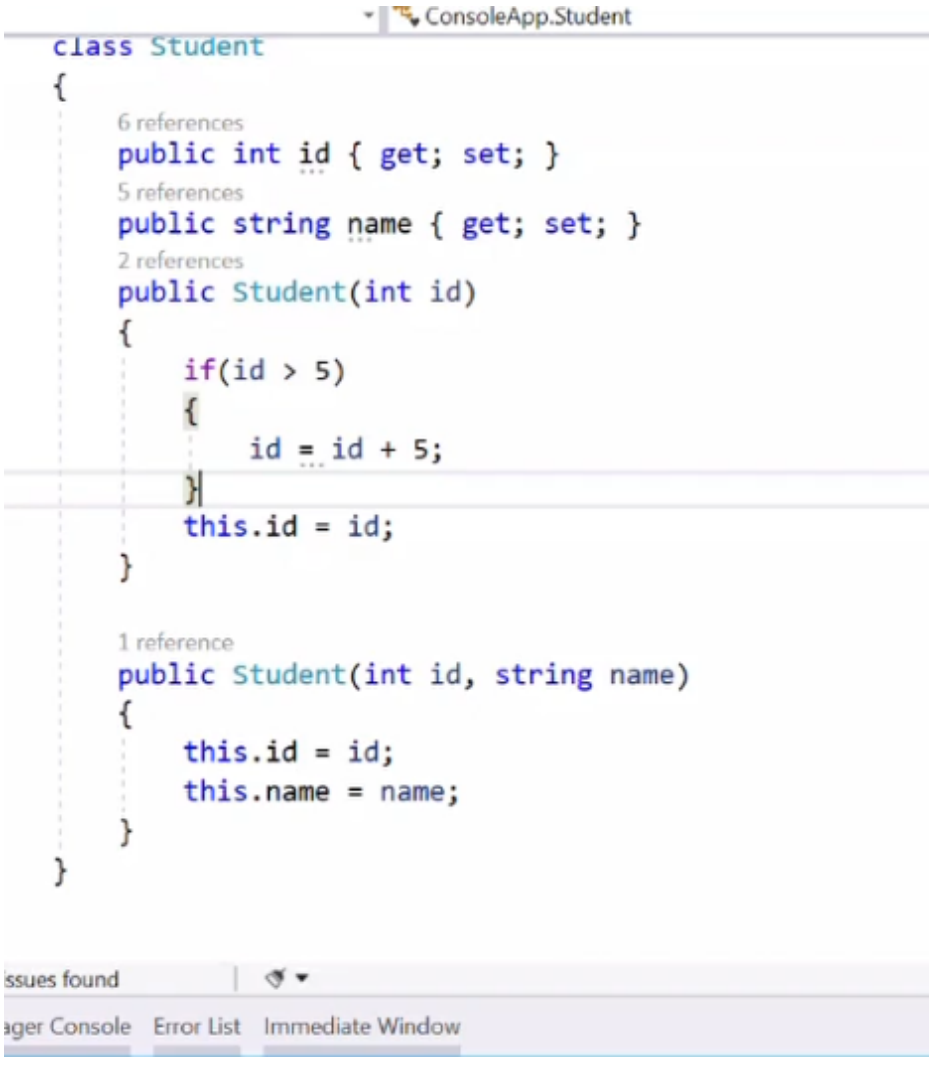
so now i'm passing id as well as name



look at this we have'nt pass name in the above console so it will be initialized to null and null is not garbage

Now i want to put a condition here

if user passes a id and the id is greater than 5 then i should say id = id + 5; i want to add +5 to it and i want to store it


ConsoleApp.Student
class Student
{
6 references
public int id { get; set; }
5 references
public string name { get; set; }
2 references
public Student(int id)
{
if(id > 5)
{
id = id + 5;
}
this.id = id;
}
1 reference
public Student(int id, string name)
{
this.id = id;
this.name = name;
}
}
ssues found
ager Console
Error List
Immediate Window