

C#-6

StringDemo1.cs* StringDemo.cs ConsoleApp*

mscorlib

```
1  +[Assembly mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089]
4
5  +using ...
10
11 -namespace System.Text
12 {
13     ...public sealed class StringBuilder : ISerializable
14     {
15         ...
16         ...public StringBuilder();
17         ...public StringBuilder(int capacity);
18         ...public StringBuilder(string value); you can directly pass a string to your String builder which will get initialised at the beginning
19         ...public StringBuilder(string value, int capacity);
20         ...public StringBuilder(int capacity, int maxCapacity);
21         ...public StringBuilder(string value, int startIndex, int length, int capacity);
22
23         ...
24         ...public char this[int index] ...
25         ...public int MaxCapacity { get; }
26         ...public int Length { get; set; }
27         ...public int Capacity { get; set; }
28
29         ...
30         ...public StringBuilder Append(object value);
31         ...public StringBuilder Append(char[] value);
32         ...public StringBuilder Append(ulong value);
33         ...public StringBuilder Append(uint value);
34
35     }
36 }
```

121 % No issues found

Output Package Manager Console Error List Immediate Window

you can see it has 6 overloads

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

StringDemo1.cs X StringDemo.cs ConsoleApp*

ConsoleApp

ConsoleApp.StringDemo1

```
7  namespace ConsoleApp
8  {
9      class StringDemo1
10     {
11         static void Main()
12         {
13             StringBuilder readData = new StringBuilder();
14             Console.WriteLine("Enter document. Press '0' to exit:");
15             string data = string.Empty;
16             do
17             {
18                 data = Console.ReadLine();
19                 if(data == "0")
20                 {
21                     break;
22                 }
23                 readData = readData.Append(data);
24             } while (1==2);
25         }
26     }
27 }
28
```

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



Class

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 stimulus, it may be like a mosquito coming 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

Should
Have



Class

Has defined
state &
behaviour



Object

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

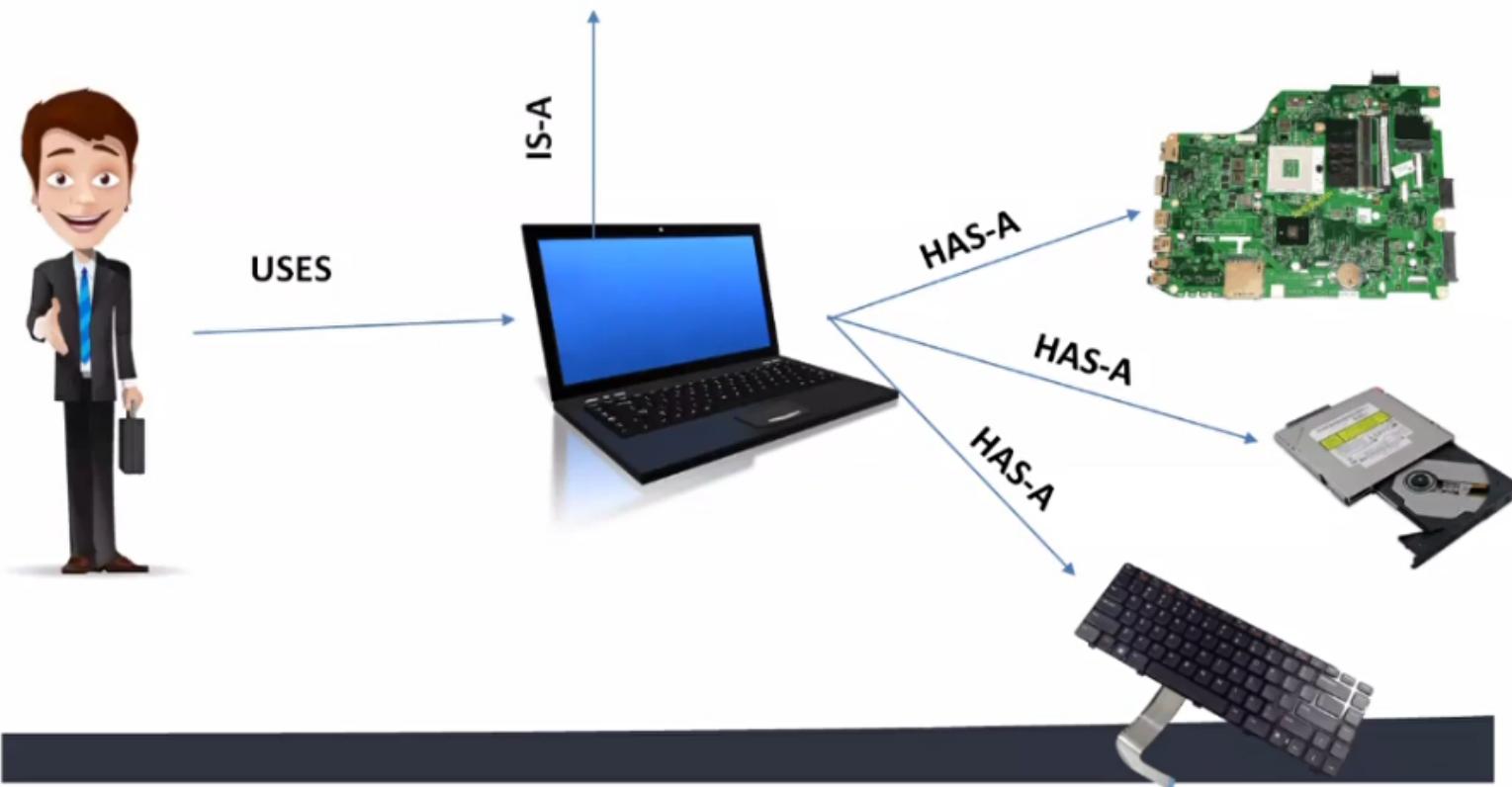
1. **Is a**

2. **Has a**

3. **Uses**

Interaction of the Objects

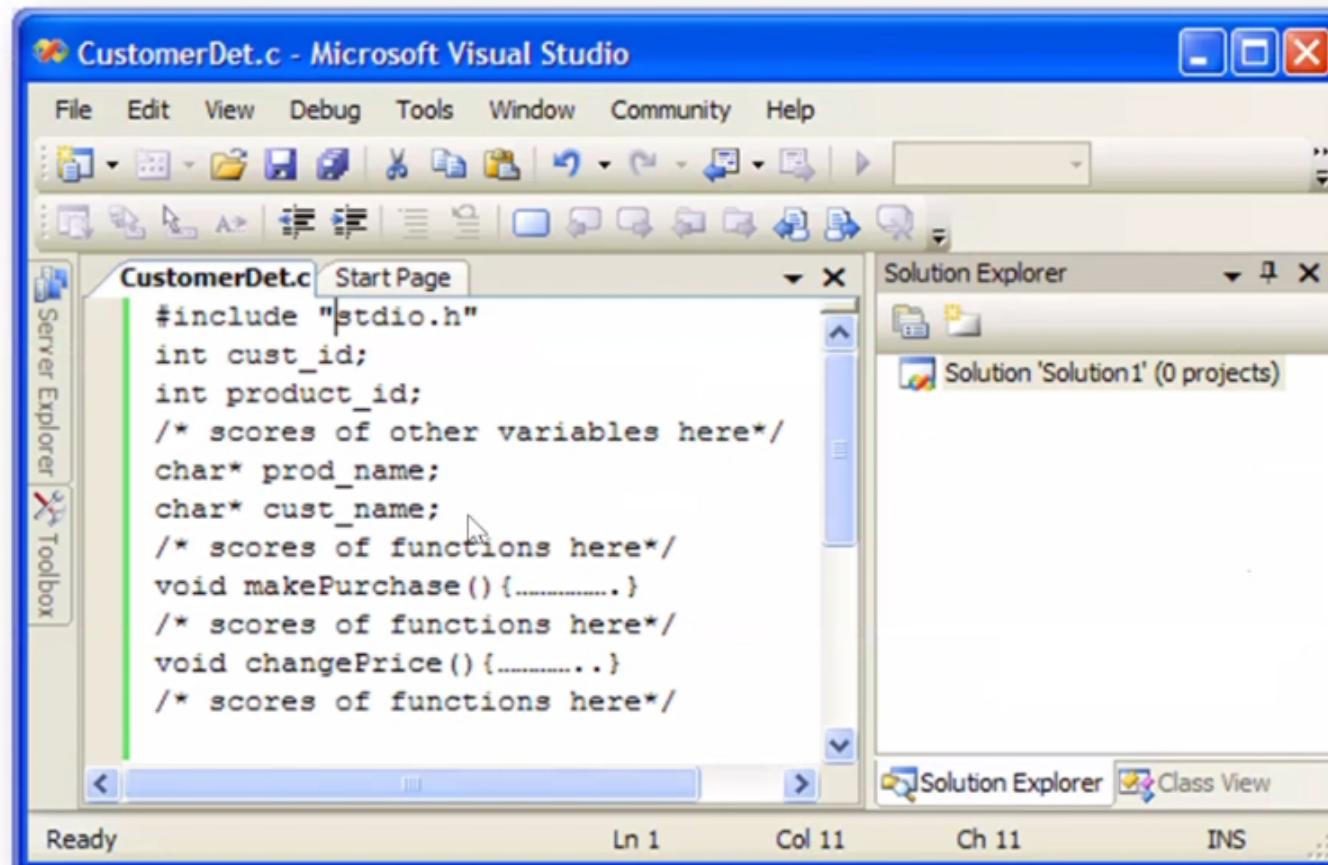
Electronic Gauged



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?

The screenshot shows the Microsoft Visual Studio IDE interface with a C file named "CustomerDet.c" open. The code includes declarations for variables like cust_id, product_id, and pointers for prod_name and cust_name, along with function prototypes for makePurchase and changePrice. Two dark blue thought bubbles are overlaid on the right side of the code editor. The top bubble contains the text "Can you identify where is Customer ?" and the bottom bubble contains "Can you identify where is Product ?".

```
#include <stdio.h>
int cust_id;
int product_id;
/* scores of other variables here*/
char* prod_name;
char* cust_name;
/* scores of functions here*/
void makePurchase(){.....}
/* scores of functions here*/
void changePrice(){.....}
/* scores of functions here*/
```

So, if 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 ?

```
Customer.java X
public class Customer {
    int id;
    String name;

    public void makePurchase()
    {
        // .....
    }
}

Product.java X
public class Product {
    int id;
    String name;

    public void changePrice()
    {
        // .....
    }
}
```

Easier to
comprehend

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 ▪ Your car
 - An order ▪ 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 straightforward 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

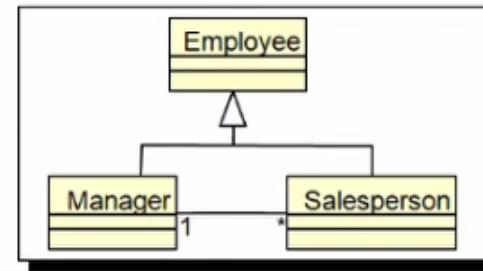
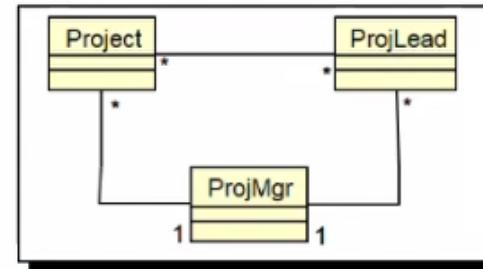
When you holding a remote, so are you bothered about what is hidden to or you bothered about what is visible to you, so when i am holding a remote i'm bothered more about what is visible to me which i can see, when i press this button what should happen i should know this,

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 interested 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

Employee
• empId : String • name : String • address : Address
◆ getEmpID() : String ◆ setEmpId(empId : String) ◆ getName() : String ◆ setName(name : String) ◆ getAddress() : Address ◆ setAddress(address : Address)

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>>

Generalization
Realization

<<Has-a>>

Association
Aggregation
Composition

<<Uses>>

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

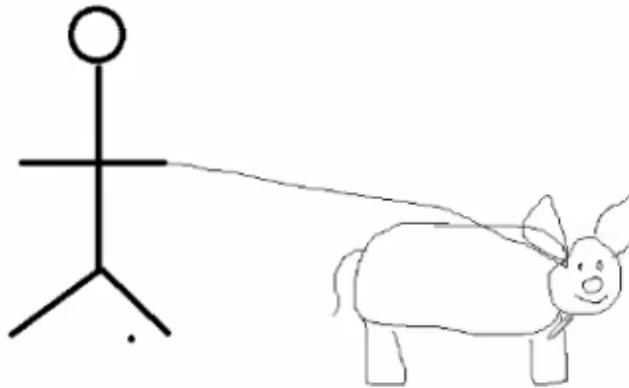
for example How many refill a pen will have

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



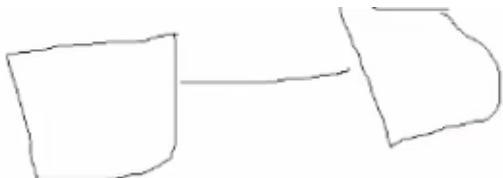
this says one way relationship

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**

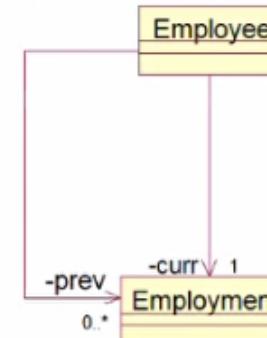
So to represent bi directional you should have to draw a line

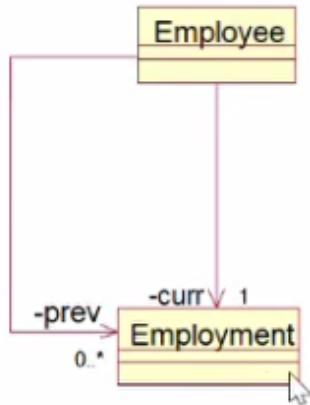


3. **Role Name** In the UML whenever you see this diagram you can always name the object whatever you feel like you can call the object as a, b, cwhatever
But whenever you see a role name you have to create a object with the same name as what we have in the role name its compulsory

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





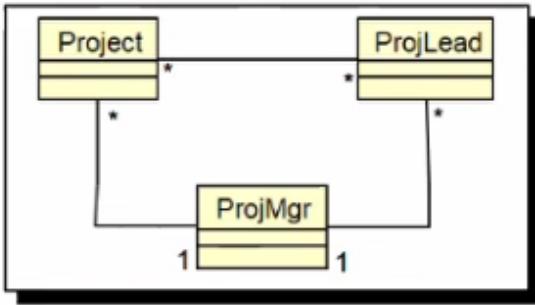
Now look at this i have a employee class and i have a employment

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?



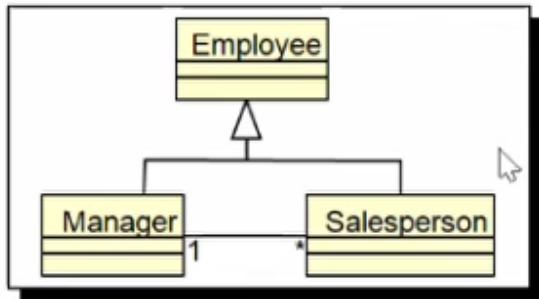
This clearly says 1 project manager has many project lead
 1 project manager has many projects to handle
 and every project will have at least 1 manager
 its bi directional project will have project manager information and project manager
 will have project information
 here also it is bi directional 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



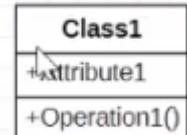
Here Manager is a employee
salesperson is a employee

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')

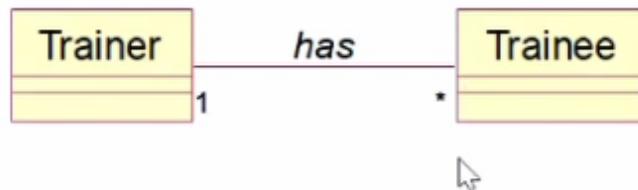
- Trainer
- Trainee
- Course
- Technology
- Module
- Unit
- Topic

Identifying Classes

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

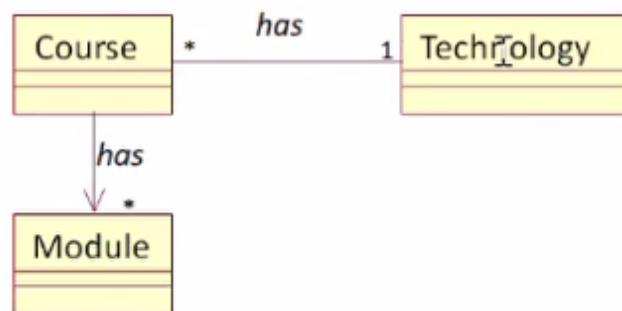
Identifying Relationships

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



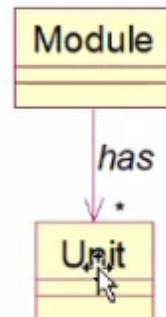
Identifying Relationships

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

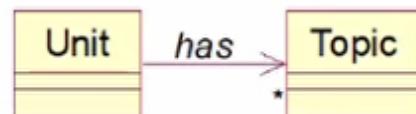


Identifying Relationships

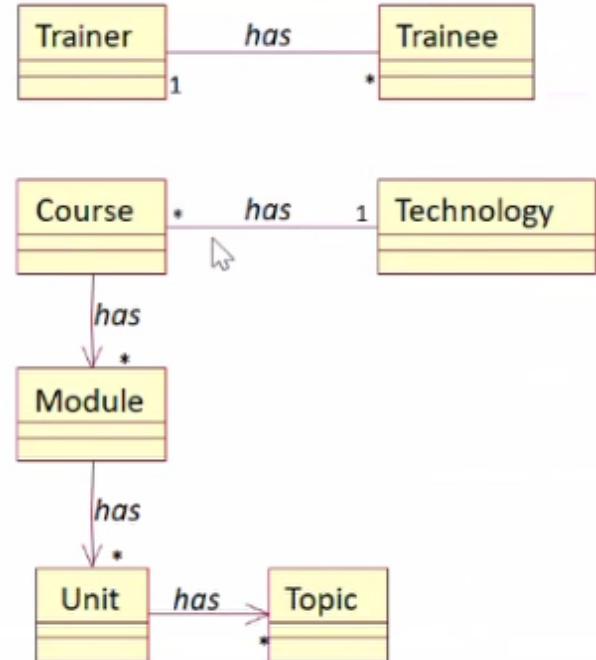
- Module – Unit
 - Module ‘HAS’ many Units



- Unit – Topic
 - Unit ‘HAS’ many Topics



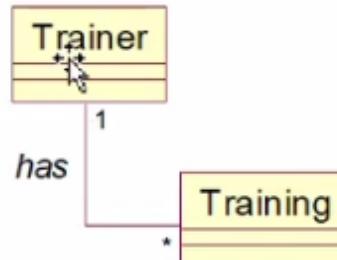
The OO Model



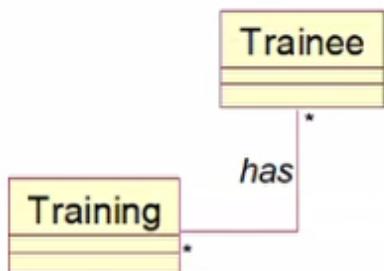
- How do you relate the Trainer & Trainee to the Course?

Conceptual Entity

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

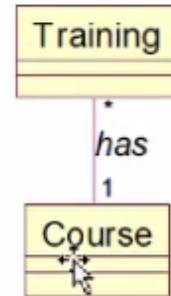


- Trainee – Training
 - A Trainee (HAS) attends many Trainings
 - A Training HAS a many Trainees

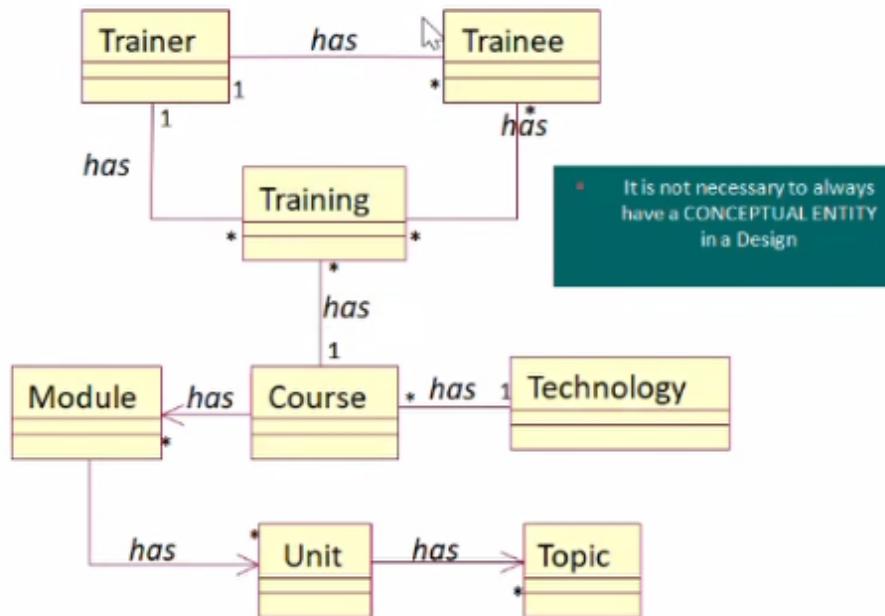


Conceptual Entity

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

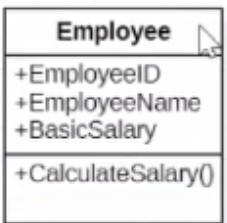


Solution



- Easier to model real-world problems through the OO approach than through the procedural approach

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* X Program.cs*

C# OODemos

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace OODemos
8  {
9      class Employee
10     {
11         // i can make this public
12         public int empId;
13         public string employeeName;
14         public double empSalary;
15         //all these are my fields
16     }
17 }
18
```

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

```
Program.cs ✘ X
mos
    ▾ OODemos.Program
        ▾ DisplayEmployee(Employee employee)

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace OODemos
{
    class Program
    {
        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
        private static void DisplayEmployee(Employee employee)
        {
            Console.WriteLine("Id = {0}\nName = {1}\nBasic Salary = {2}", employee.empId, employee.employeeName, employee.empSalary);
        }
    }
}
```

What's 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* X Program.cs

C# OODemos

```
7  namespace OODemos
8  {
9      class Employee
10     {
11         //fields
12         int empId;
13         string employeeName;
14         double salary;
15
16         //Now who can access them
17         //i should have a member of the house who can access the information
18         // Now if a external person want to access this data so it can access only through you,
19         // if you are not there i cannot access data
20         //so if i want to access the jewellery i should be the part of the family
21         //so you should make particular class as the part of the class, How do you do that
22
23         //In Programming term we called it as----- INHERITANCE-----
24         //We'll Extend that particular class
25         //so after extending he can access the jewellery
26         //again we can set set rules and regulations so that he cannot access all the jewellery
27         //like in c++ friend function can access private data
28         0 references
29         int GetEmpId()
30         {
31             //who will return me the employee id
32             return empId;
33         }
34     }
35 }
```

124 % No issues found

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

Program.cs X

OODemos.Program

Main(string[] args)

```
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace OODemos
{
    class Program
    {
        static void Main(string[] args)
        {
            Employee employee = new Employee();
            employee.empId = 1001;
            employee.empName = "John Doe";
            employee.empSalary = 10000;
            //to display the details
            DisplayEmployee(employee);
        }
        //in display method we just take the employee object and we'll print them
        private static void DisplayEmployee(Employee employee)
        {
            Console.WriteLine("Id = {0}\nName = {1}\nBasic Salary = {2}", employee.empId, employee.empName, employee.empSalary);
        }
    }
}
```

Employee.empId : 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* ✘ X Program.cs*

c# OODemos

```
9     class Employee
10    {
11        //fields
12        int empId;
13        string employeeName;
14
15        0 references
16        public int GetEmpId()
17        {
18            return empId;
19        }
19        0 references
20        public void SetEmpId(int empId)
21        {
22            empId = empId;
23        }
23        0 references
24        public string GetEmployeeName()
25        {
26            return employeeName;
27        }
27        0 references
28        public void SetEmployeeName(string employeeName)
29        {
30            employeeName = employeeName;
30        }
30    }
```

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

Program.cs x x

OODemos.Program Main(s)

```
namespace OODemos
{
    class Program
    {
        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");

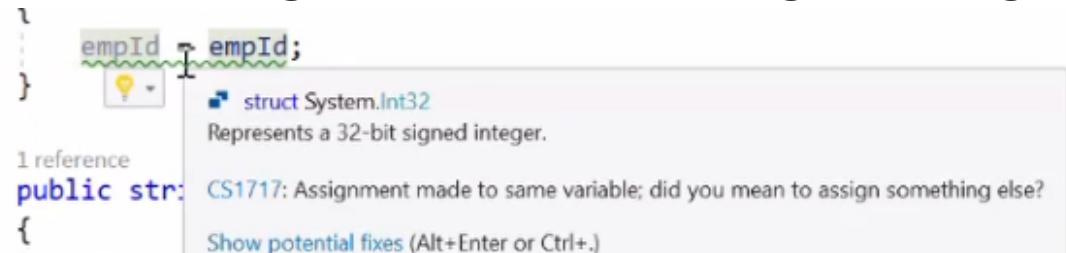
            DisplayEmployee(employee);
        }

        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 highlighted
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; ← 2. then i want to assign this to
    string employeeName; objects empid
    //double empSalary;

    1 reference
    public int GetEmpId()
    {
        return empId;
    }
    1 reference
    public void SetEmpId(int empId) ← 1. i want to take the
    {
        empId = empId; empid from here and
    } 3. so how do you say that this
        particular empid belongs to the object

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

```

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

```
this.empId = empId;
class OODemos.Employee
```

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

now is this a good way
of writing

```
1 reference
public string GetEmpName()
{
    return employeeName;
}
1 reference
public void SetEmpName(string empName)
{
    employeeName = empName;
}
```

or this is a good way
of writing

what is the difference

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

but in 2nd case

```
//variables
int empId;
string employeeName; ← here you have a
//double empSalary;

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

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

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

1 reference
public void SetEmpName(string empName)
{
    employeeName = empName;
}
```

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 that's a reason this particular variable name should be same

```
1 reference
public void SetEmpName(string employeeName)
{
    employeeName = employeeName;
}
```

Now as this is same you can see when i double click on any one all this becomes highlighted (local)

so you should always say boss this belongs to the object, so you should always say this.

```
1 reference
public void SetEmpName(string employeeName)
{
    this.employeeName = employeeName;
}
```

2nd What is the problem in this particular program ?

The screenshot shows a Microsoft Visual Studio interface with two tabs open: 'Employee.cs' and 'Program.cs'. The 'Employee.cs' tab is active, displaying the following C# code:

```
Employee.cs  X  Program.cs
OODemos
class Employee
{
    //fields
    int empId;
    string employeeName;
    //double empSalary;

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

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

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

    1 reference
    public void SetEmpName(string employeeName)
    {
        this.employeeName = employeeName;
    }
}
```

Now imagine this is a simple program and you have 2 fields and for this 2 fields you are writing sooo 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

```
int x;  
x = 50; //  
int y = x;
```

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;
```

DifferentEmployee(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

```
0 references
public string EmployeeName
{
    set
    {
        this.employeeName = value;
    }
    get
    {
        return this.employeeName;
    }
}
```

```
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 X

OODemos

```
7  namespace OODemos
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             Employee employee = new Employee();
14             employee.SetEmpId(1001);
15             employee.SetEmpName("Shashi");
16             //employee.empSalary = 230000;
17
18             DisplayEmployee(employee);
19         }
20
21         private static void DisplayEmployee(Employee employee)
22         {
23             Console.WriteLine("Id = {0}\nName = {1}",
24                             employee.GetEmpId(), employee.GetEmpName());
25         }
}
```

133 % 4 0

Output Package Manager Console Error List Immediate Window

Now here i'm getting the error
so i'm doing the same thing as
we initialize the variable

```
namespace OODemos
{
    class Program
    {
        static void Main(string[] args)
        {
            Employee employee = new Employee();
            employee.EmployeeId = 1001;
            employee.EmployeeName = "Shashi";
            //employee.empSalary = 230000;

            DisplayEmployee(employee);
        }

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

No issues found

Manager Console Error List ... Immediate Window

so we do like this
look at this, this is soo
simple

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* X Program.cs

OODemos

```
9      class Employee
10     {
11         //fields
12         string empId;
13         string employeeName;
14         //double empSalary;
15
16         2 references public string EmployeeId
17         {
18             set
19             {
20                 this.empId = "E"+value;
21             }
22             get
23             {
24                 return this.empId;
25             }
26         }
27
28         2 references public string EmployeeName...
```

for that i'll take empid as string

and concatenate it with E

133 % No issues found

Employee.cs Program.cs X

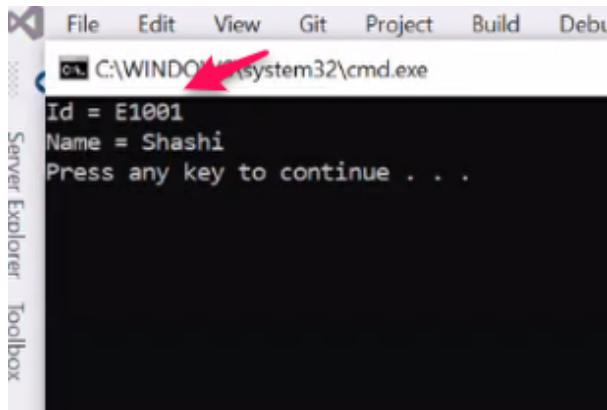
OODemos

```
7  namespace OODemos
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             Employee employee = new Employee();
14             employee.EmployeeId = "1001"; ← now you can see i have passed
15             employee.EmployeeName = "Shashi";
16             //employee.empSalary = 230000;
17
18             DisplayEmployee(employee);
19         }
20
21         private static void DisplayEmployee(Employee employee)
22         {
23             Console.WriteLine("Id = {0}\nName = {1}",
24                             employee.EmployeeId, employee.EmployeeName);
25         }
}
```

133 % No issues found

Output Package Manager Console Error List ... Immediate Window

now you can see i have passed
employeeid a 1001
but when i fetch it i'll get E1001



So i can manipulate 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 manipulate the data if it is a straight forward code

Employee.cs* X Program.cs

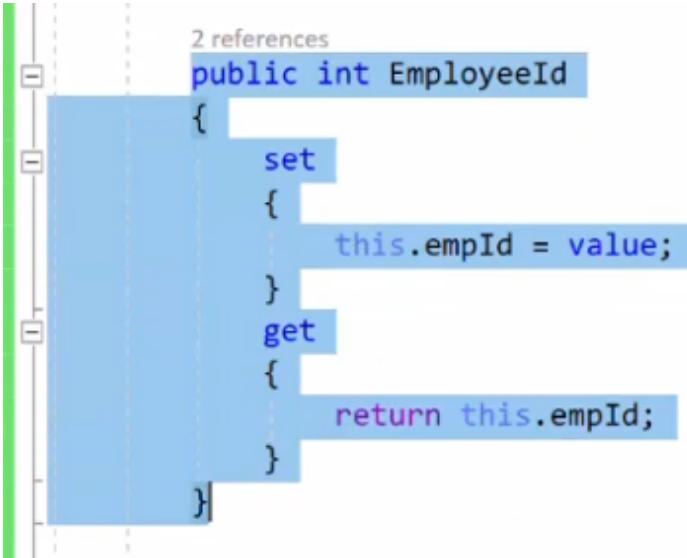
OODemos OODemos.Employee

```
9     class Employee
10    {
11        //fields
12        int empId;
13        string employeeName;
14        //double empSalary;
15
16        2 references
17        public int EmployeeId
18        {
19            set
20            {
21                this.empId = value;
22            }
23            get
24            {
25                return this.empId;
26            }
27
28        2 references
29        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**

```
public int MyProperty { get; set; }
```

and edit accordingly

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

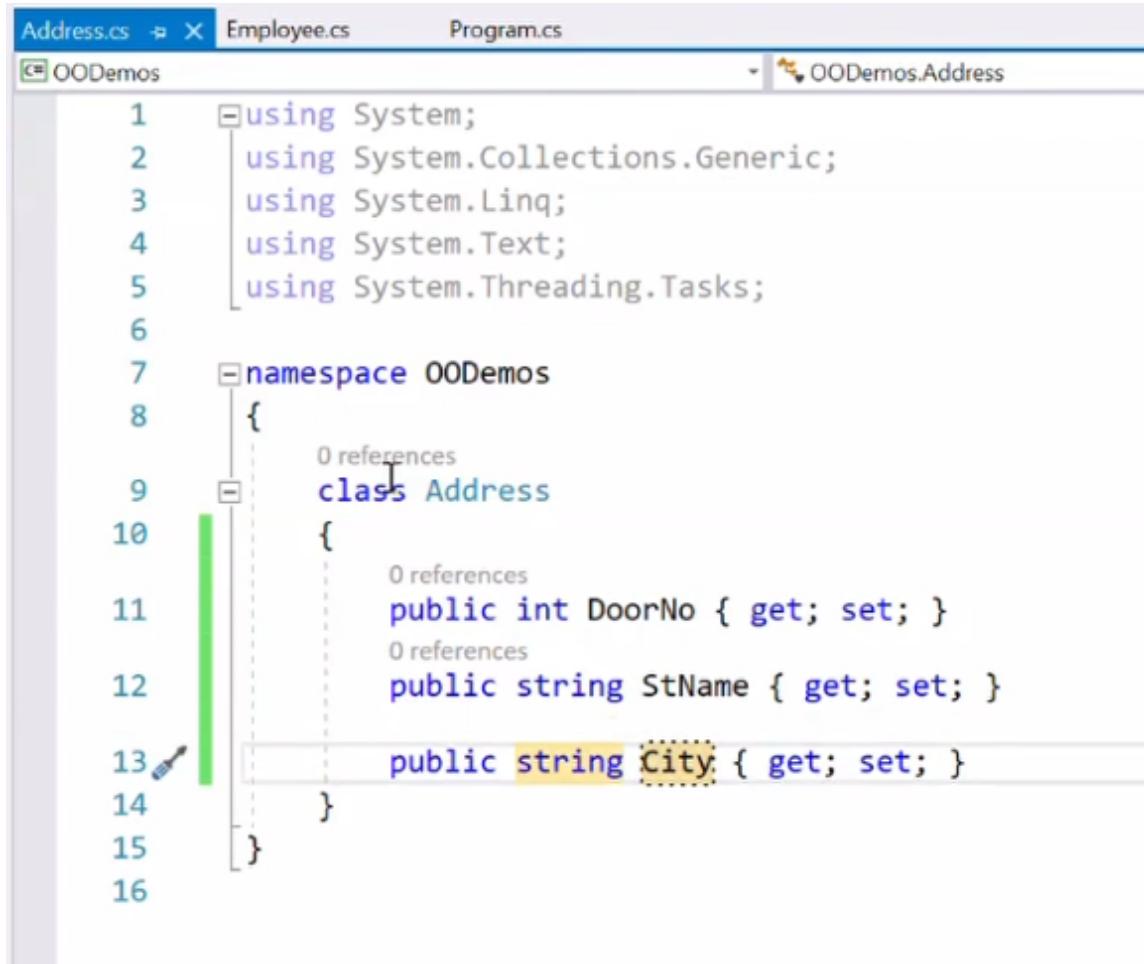
that's it, look how easy this is,

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



The screenshot shows a code editor with three tabs at the top: Address.cs, Employee.cs, and Program.cs. The Address.cs tab is selected. The code editor displays the following C# code:

```
Address.cs  X Employee.cs  Program.cs
OODemos.Address

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace OODemos
8  {
9      class Address
10     {
11         public int DoorNo { get; set; }
12         public string StName { get; set; }
13         public string City { get; set; }
14     }
15 }
16
```

A green vertical bar highlights the entire class definition from line 9 to line 14. A cursor is positioned on the word "City" in the line "public string City { get; set; }". The status bar at the bottom of the code editor shows the path "OODemos.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

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

so just like empid empname we have empaddress

```
Employee.cs Program.cs X
OODemos.Program
class Program
{
    0 references
    static void Main(string[] args)
    {
        //create employee
        Employee employee = new Employee();
        employee.EmployeeId = 1001;
        employee.EmployeeName = "Shashi";

        //create address
        Address address = new Address();
        address.City = "Bangalore";
        address.DoorNo = 71;
        address.StName = "Hulimavu";

        //add address to employee
        employee.Address = address;
        DisplayEmployee(employee);
    }
}
```

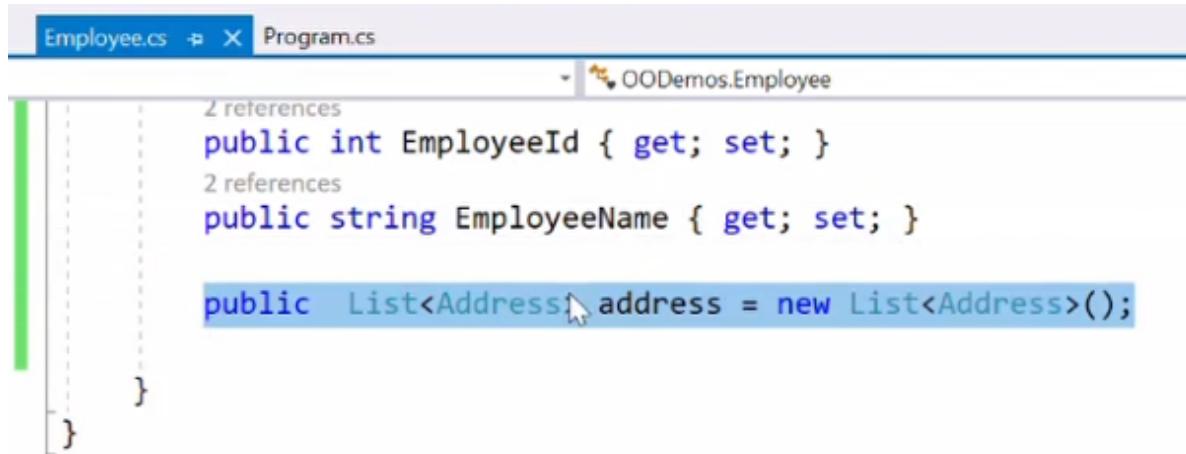
this is how we fill data in address object

and this is how we pass this address object to
this employee

Now this is for one address but see in amazon we can give 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



```
Employee.cs  X Program.cs
OODemos.Employee
2 references
public int EmployeeId { get; set; }
2 references
public string EmployeeName { get; set; }

public List<Address> address = new List<Address>();
}

}
```

now see in program you cannot add address directly

The screenshot shows a Microsoft Visual Studio code editor window titled "Program.cs". The code is written in C# and demonstrates object-oriented programming concepts. It creates an Employee object, sets its ID and name, creates an Address object, and then associates the Address with the Employee. Finally, it calls a method to display the Employee information.

```
Program.cs  X
OODemos.Program
static void Main(string[] args)
{
    //create employee
    Employee employee = new Employee();
    employee.EmployeeId = 1001;
    employee.EmployeeName = "Shashi";

    //create address
    Address address = new Address();
    address.City = "Bangalore";
    address.DoorNo = 71;
    address.StName = "Hulimavu";

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

    DisplayEmployee(employee);
}

1 reference
```

look at this when i say employee.address

```
//add address to employee  
employee.add_  
employee. + address (field) List<Address> Employee.address  
+ - x  
DisplayEmployee(employee);
```

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 ?

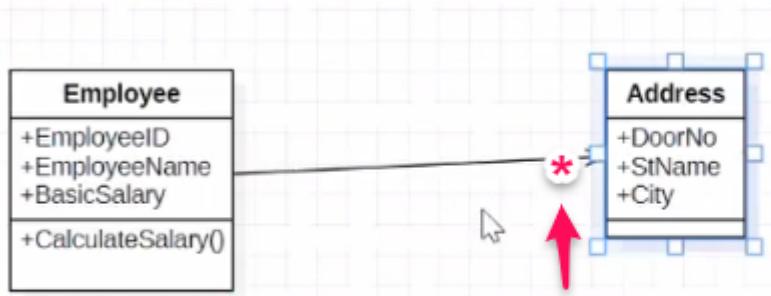
so who develop the employee class it is his responsiblity that it stores 1 address or 100 address that is not my problem

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,



when ever there is a star here

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>(); I

}
```

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

The screenshot shows a code editor window with two tabs: "Employee.cs" and "Program.cs*". The "Employee.cs" tab is active, displaying the following C# code:

```
Employee.cs  X Program.cs*
OODemos.Employee

2 references
public int EmployeeId { get; set; }

2 references
public string EmployeeName { get; set; }

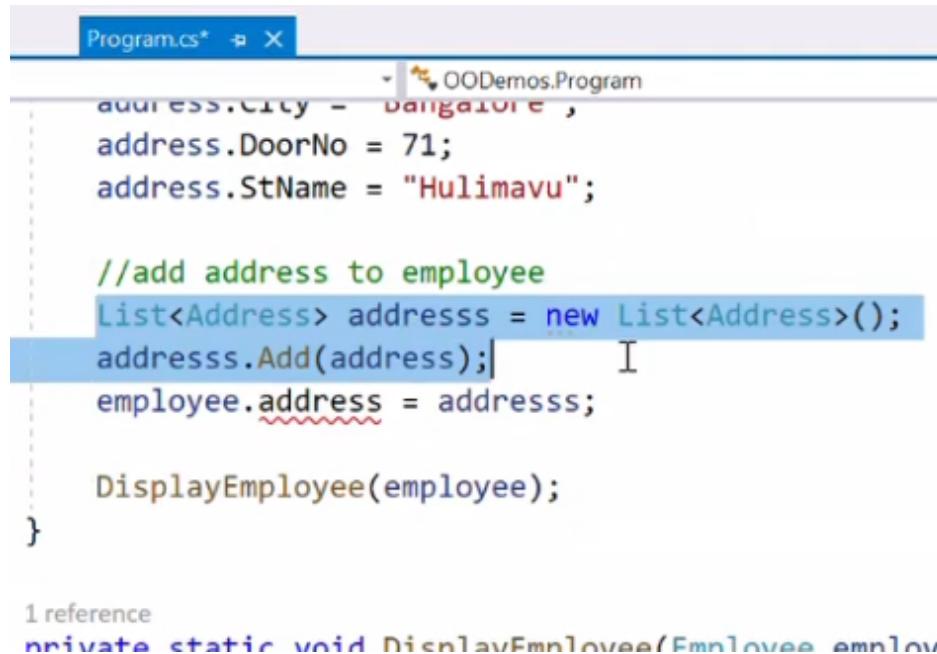
private List<Address> address = new List<Address>();

0 references
public void AddAddress(Address address)
{
    this.address.Add(address);
}

0 references
public List<Address> GetAddresses()
{
    return this.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



The screenshot shows a Microsoft Visual Studio code editor window titled "Program.cs". The code is part of a project named "OODemos.Program". The code defines an Employee object and adds an address to it:

```
address.City = "Bangalore",
address.DoorNo = 71;
address.StName = "Hulimavu";

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

DisplayEmployee(employee);
}

1 reference
private static void DisplayEmployee(Employee employee)
```

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

```
Program.cs* X
OODemos.Program
address.City = "Bangalore",
address.DoorNo = 71;
address.StName = "Hulimavu";

//add address to employee

employee.AddAddress(address);

DisplayEmployee(employee);
}
```

Now how will i access the address data ?

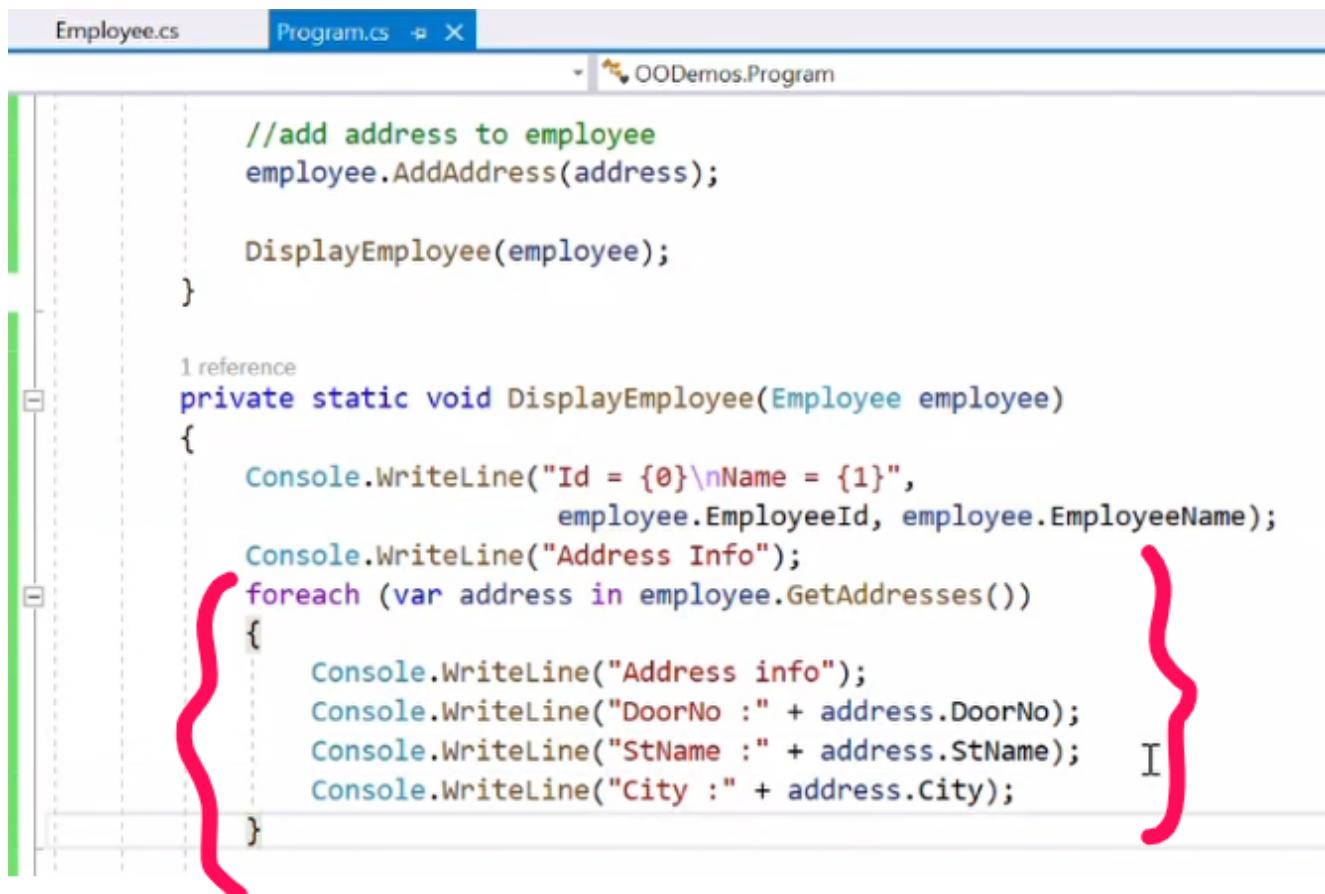
```
1 reference
private static void DisplayEmployee(Employee employee)
{
    Console.WriteLine("Id = {0}\nName = {1}",
                      employee.EmployeeId, employee.EmployeeName);

    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
employee 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**



```
Employee.cs      Program.cs  X
OODemos.Program

    //add address to employee
    employee.AddAddress(address);

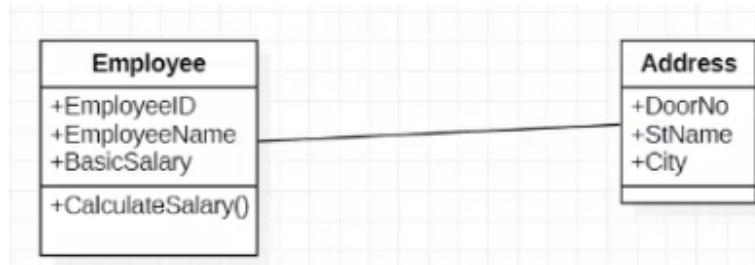
    DisplayEmployee(employee);
}

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

So many means i have to create list, I can create any data structure whatever we have in the library 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

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar displays "Employee.cs" and "Program.cs". Below the title bar, the solution name "OODemos.Employee" is visible. The main code editor area contains the following C# code:

```
private List<Address> address = new List<Address>();  
1 reference  
public void AddAddress(Address address)  
{  
    this.address.Add(address);  
}  
1 reference  
public List<Address> GetAddresses()  
{  
    return this.address;  
}  
}
```

i'll say employee has a address

The screenshot shows a code editor with two tabs: "Employee.cs" and "Program.cs". The "Employee.cs" tab is active, displaying the following C# code:

```
OODemos.Employee
    this.employeeName = value;
}
get
{
    return this.employeeName;
}
*/
2 references
public int EmployeeId { get; set; }
2 references
public string EmployeeName { get; set; }

public Address Address { get; set; }
}
```

in the same way my address will also has the employee

Address.cs Employee.cs Program.cs

OODemos OODemos.Address

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace OODemos
8  {
9      class Address
10     {
11         public int DoorNo { get; set; }
12         public string StName { get; set; }
13         public string City { get; set; }
14         public Employee Employee { get; set; }
15     }
16 }
17
18
```

```
Address.cs Employee.cs Program.cs* + X OODemos.OODemos.Program
11
12 {
13     //create employee
14     Employee employee = new Employee();
15     employee.EmployeeId = 1001;
16     employee.EmployeeName = "Shashi";
17
18     //create address
19     Address address = new Address();
20     address.City = "Bangalore";
21     address.DoorNo = 71;
22     address.StName = "Hulimavu";
23
24     //add address to employee ←
25     employee.Address = address;
26
27     //add employee to address ← I
28
29     DisplayEmployee(employee);
30 }
31
32 private static void DisplayEmployee(Employee employee)
33 {
```

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

```
Address.cs  X Employee.cs      Program.cs
OODemos.Address
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace OODemos
8  {
9      class Address
10     {
11         public int DoorNo { get; set; }
12         public string StName { get; set; }
13         public string City { get; set; }
14         public List<Employee> Employees { get; set; }
15     }
16 }
17
```

if one address has list of employee
then this will be a private list
and we'll have a add and get method

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

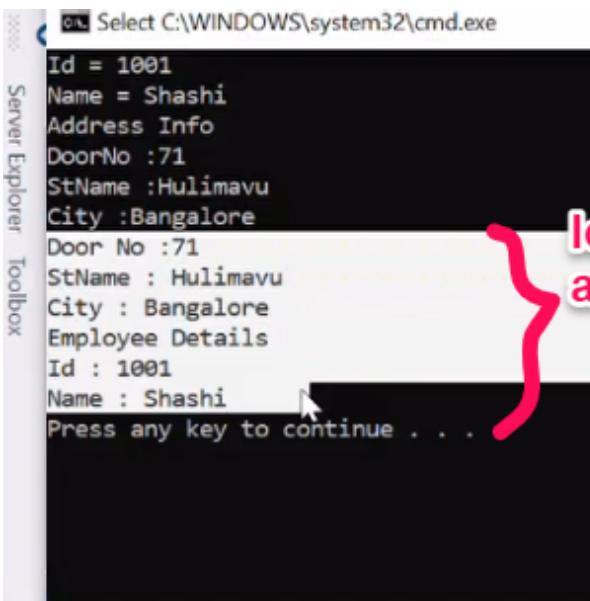
Employee.cs Program.cs X

OODemos.Program

```
        DisplayEmployee(employee);
        DisplayAddress(address);
    }

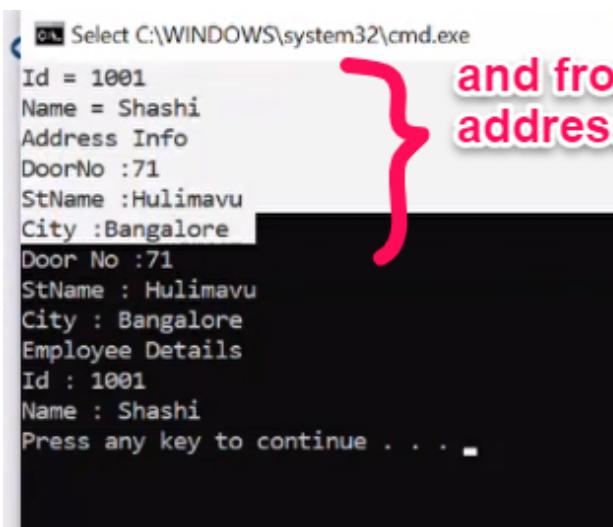
    1 reference
    private static void DisplayAddress(Address address)
    {
        Console.WriteLine("Door No :" + address.DoorNo);
        Console.WriteLine("StName : " + address.StName);
        Console.WriteLine("City : " + address.City);
        Console.WriteLine("Employee Details");
        Console.WriteLine("Id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

    1 reference
    private static void DisplayEmployee(Employee employee)...
```



```
0. 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

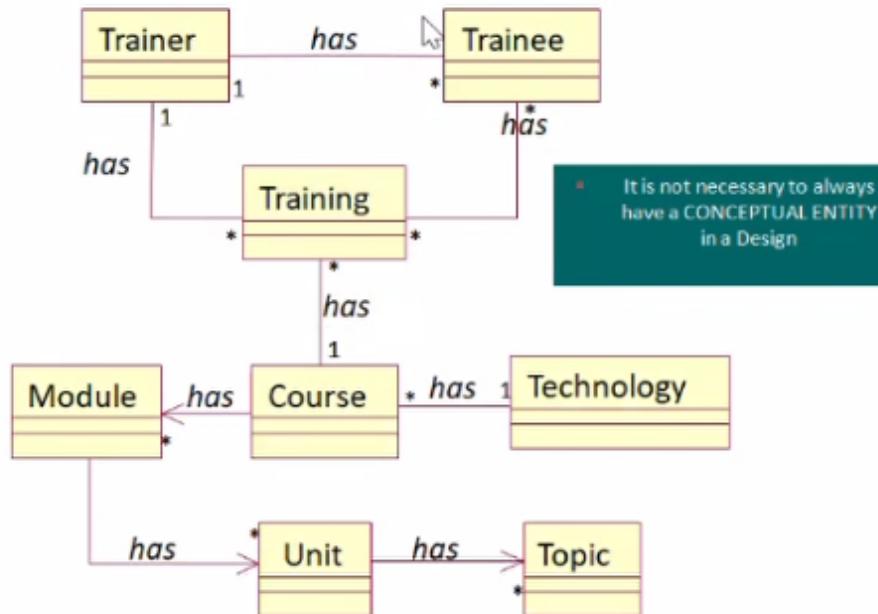


```
0. Select C:\WINDOWS\system32\cmd.exe
Id = 1001
Name = Shashi
Address Info
DoorNo :71
StName :Hulimavu
City :Bangalore
Door No :71
StName : Hulimavu
City : Bangalore
Employee Details
Id : 1001
Name : Shashi
Press any key to continue . . .
```

and from employee i'm accessing
address details

now lets code Trainer trainee

Solution

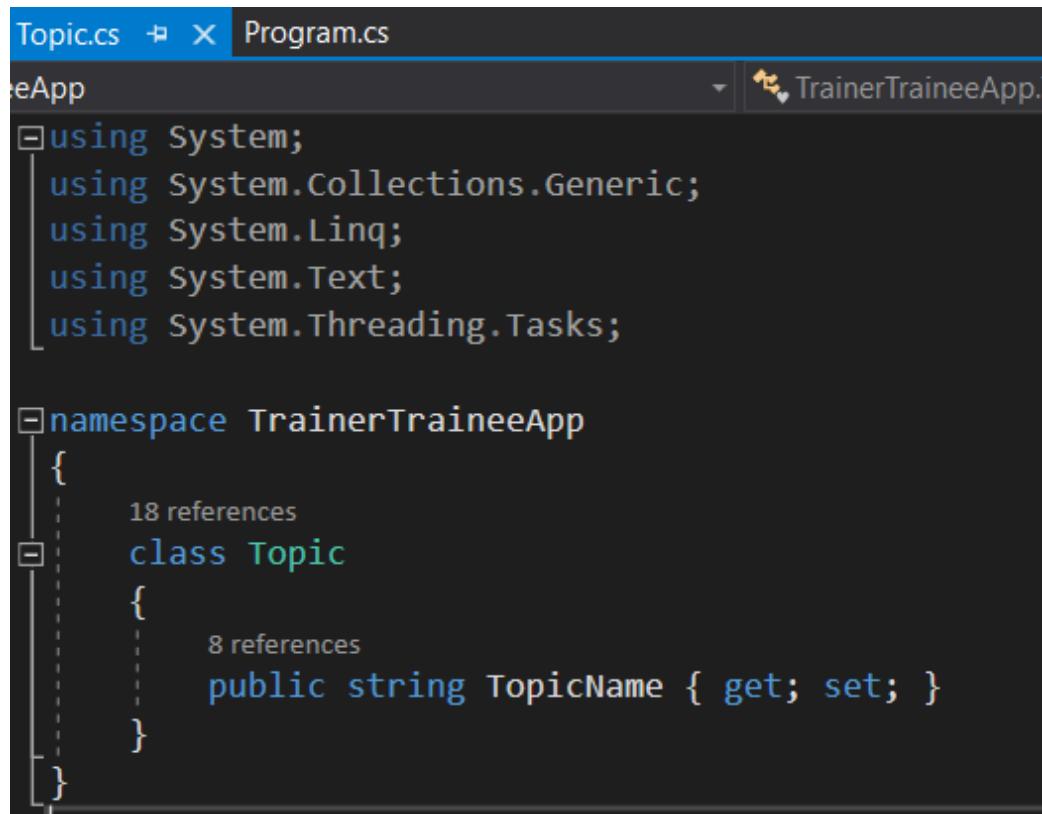


• It is not necessary to always have a CONCEPTUAL ENTITY in a Design

• Easier to model real-world problems through the OO approach than through the procedural approach

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**



The screenshot shows a code editor window with two tabs: "Topic.cs" and "Program.cs". The "Topic.cs" tab is active, displaying the following C# code:

```
Topic.cs  ✘ X Program.cs
eApp
[-] using System;
[-] using System.Collections.Generic;
[-] using System.Linq;
[-] using System.Text;
[-] using System.Threading.Tasks;

[-] namespace TrainerTraineeApp
{
    {
        18 references
        [-] class Topic
        {
            {
                8 references
                public string TopicName { get; set; }
            }
        }
    }
}
```

The code defines a class named `Topic` with a single property `TopicName` of type `string`, using auto-implemented properties (`get; set;`). The `Topic` class is located within the `TrainerTraineeApp` namespace.

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

```
Unit.cs ✘ X
C# TrainerTraineeApp TrainerTraineeApp.Unit
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace TrainerTraineeApp
8 {
9     class Unit
10    {
11        public string UnitName { get; set; }
12        private List<Topic> topics = new List<Topic>();
13
14        public void AddTopic(Topic topic)
15        {
16            this.topics.Add(topic);
17        }
18
19        public List<Topic> GetTopics()
20        {
21            return this.topics;
22        }
23    }
24 }
25
```

Now we create Module,
Module will have ModuleName, list of Units

Module.cs ✘

C# TrainerTraineeApp TrainerTraineeApp.Mod

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace TrainerTraineeApp
8 {
9     class Module
10    {
11        public string ModuleName { get; set; }
12        private List<Unit> units = new List<Unit>();
13
14        public void AddUnit(Unit unit)
15        {
16            this.units.Add(unit);
17        }
18
19        public List<Unit> GetUnits()
20        {
21            return this.units;
22        }
23    }
24 }
25
```

next we create Course

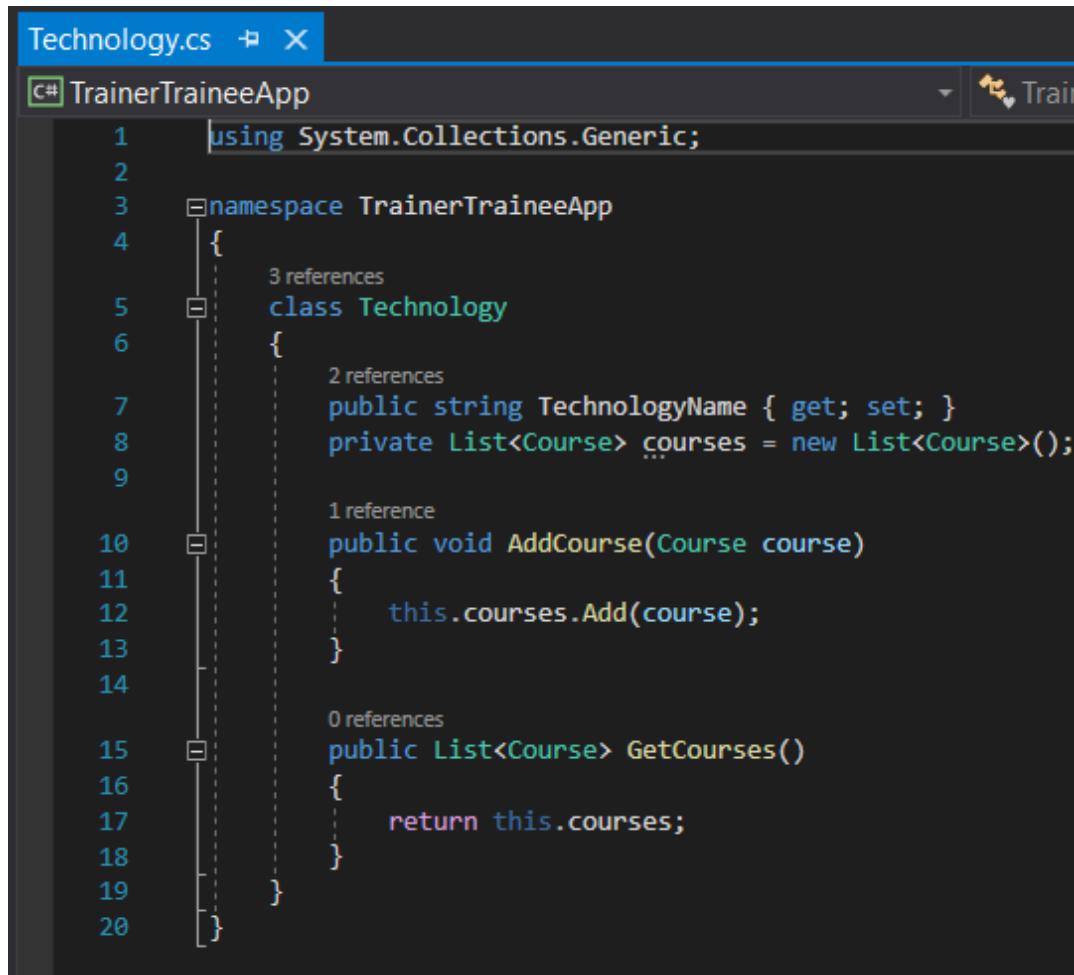
Course will have CourseName, technology, list of modules, list of trainings

Course.cs X

c# TrainerTraineeApp TrainerTrainee

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace TrainerTraineeApp
8  {
9      class Course
10     {
11         public string CourseName { get; set; }
12         public Technology Technology { get; set; }
13         private List<Module> modules = new List<Module>();
14         private List<Training> trainings = new List<Training>();
15
16         public void AddModule(Module module)
17         {
18             this.modules.Add(module);
19         }
20
21         public List<Module> GetModules()
22         {
23             return this.modules;
24         }
25
26         public void AddTraining(Training training)
27         {
28             this.trainings.Add(training);
29         }
30
31         public List<Training> GetTrainings()
32         {
33             return this.trainings;
34         }
35     }
36 }
37 }
```

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



The screenshot shows a code editor window titled "Technology.cs". The code defines a class named "Technology" with properties for "TechnologyName" and a list of "Course"s. It also contains methods for adding courses and getting the list of courses.

```
1  using System.Collections.Generic;
2
3  namespace TrainerTraineeApp
4  {
5      class Technology
6      {
7          public string TechnologyName { get; set; }
8          private List<Course> courses = new List<Course>();
9
10         public void AddCourse(Course course)
11         {
12             this.courses.Add(course);
13         }
14
15         public List<Course> GetCourses()
16         {
17             return this.courses;
18         }
19     }
20 }
```

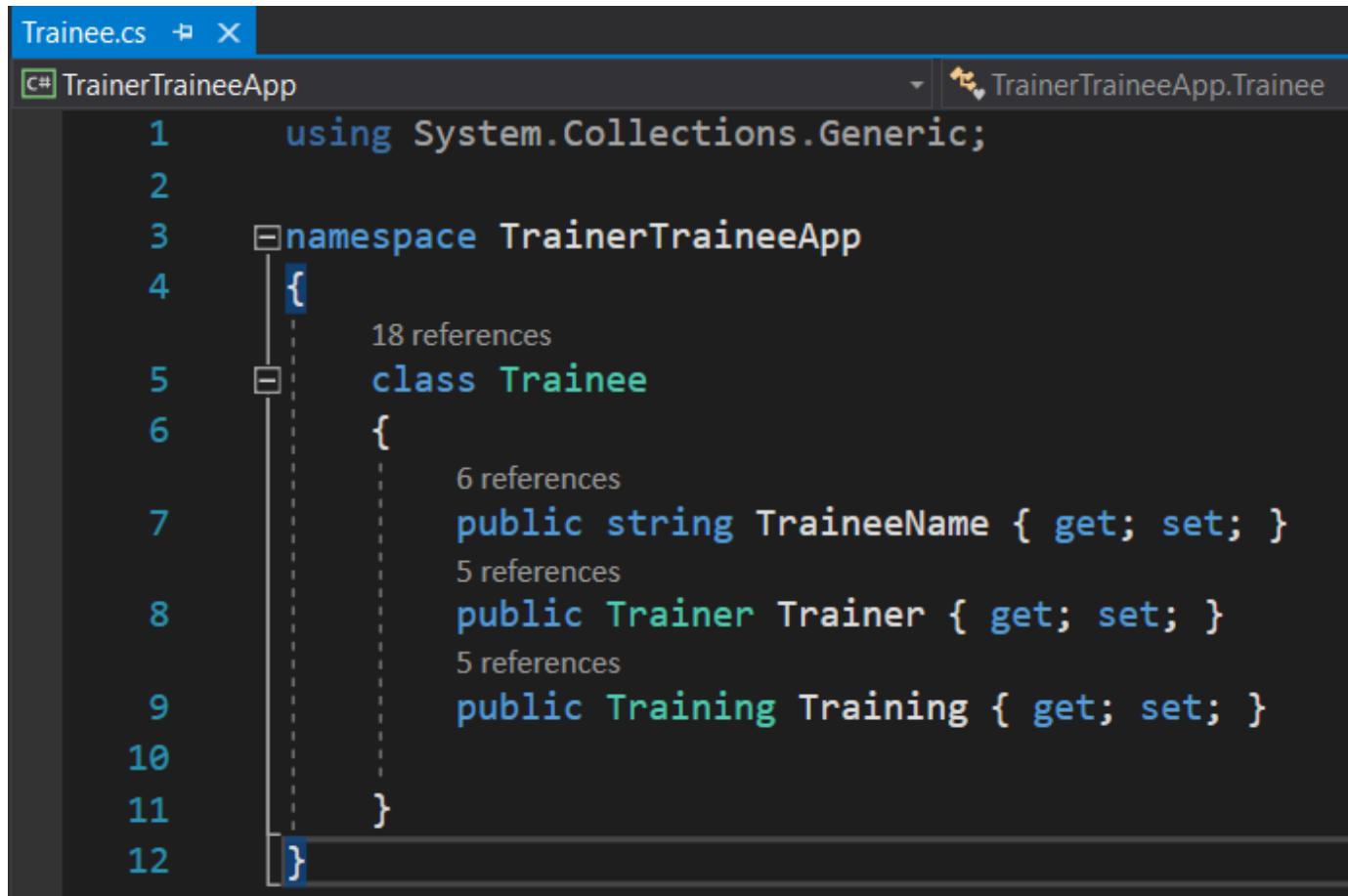
Now Training, in training we have TrainingName, Course, training has a trainer, and list or trainees

Training.cs X

C# TrainerTraineeApp TrainerTraineeApp.Training

```
1  using System.Collections.Generic;
2
3  namespace TrainerTraineeApp
4  {
5      class Training
6      {
7          public string TrainingName { get; set; }
8          public Course Course { get; set; }
9          public Trainer Trainer { get; set; }
10
11         private List<Trainee> trainees = new List<Trainee>();
12
13         public void AddTrainee(Trainee trainee)
14         {
15             this.trainees.Add(trainee);
16         }
17
18         public List<Trainee> GetTrainees()
19         {
20             return this.trainees;
21         }
22     }
23 }
```

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**



The screenshot shows a code editor window for a C# file named Trainee.cs. The file is part of a project called TrainerTraineeApp. The code defines a class Trainee with three properties: TraineeName, Trainer, and Training. The code is annotated with numbers 1 through 12 on the left side, and some parts of the code are highlighted in blue.

```
1  using System.Collections.Generic;
2
3  namespace TrainerTraineeApp
4  {
5      class Trainee
6      {
7          public string TraineeName { get; set; }
8          public Trainer Trainer { get; set; }
9          public Training Training { get; set; }
10     }
11 }
12 }
```

Now Trainer

Trainer.cs X

C# TrainerTraineeApp Trainer

```
1  using System.Collections.Generic;
2
3  namespace TrainerTraineeApp
4  {
5      class Trainer
6      {
7          public string TrainerName { get; set; }
8          public Training Training { get; set; }
9          private List<Trainee> trainees = new List<Trainee>();
10
11         public void AddTrainee(Trainee trainee)
12         {
13             this.trainees.Add(trainee);
14         }
15
16         public List<Trainee> GetTrainees()
17         {
18             return this.trainees;
19         }
20     }
21 }
```

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

HTML

p tag
a tag
table tag

JS

JS Intro
JS Loops

CSS

Fonts
Borders

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 ✘ X
C# TrainerTraineeApp TrainerTraineeApp.Program
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace TrainerTraineeApp
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            //1. create topic
14            Topic topic1 = new Topic();
15            topic1.TopicName = "p tag";
16            Topic topic2 = new Topic() { TopicName = "a tag" };
17            Topic topic3 = new Topic() { TopicName = "table tag" };
18            Topic topic4 = new Topic() { TopicName = "JS Intro" };
19            Topic topic5 = new Topic() { TopicName = "JS Loops" };
20            Topic topic6 = new Topic() { TopicName = "Fonts" };
21            Topic topic7 = new Topic() { TopicName = "Borders" };
22        }
23    }
}
```

```
//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
151     □
152     □
153     □
154     {
155         for (int i = 0; i < 35; i++)
156         {
157             Console.WriteLine("-");
158         }
159     }
160 }
161
```

output

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

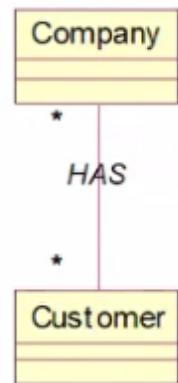
Identifying Classes

- Company
- Item
- Order
- Customer
- RegCustomer



Identifying Relationships

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



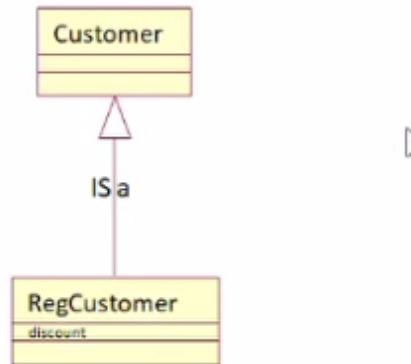
Identifying Relationships

- Company - Item
 - Company HAS many Items



Identifying Relationships

- Customer - RegCustomer
 - RegCustomer 'IS' a Customer



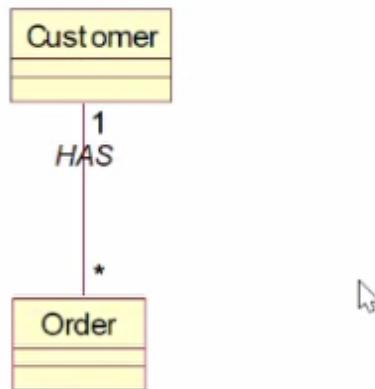
Identifying Relationships

- Order - Item
 - Order HAS many Items

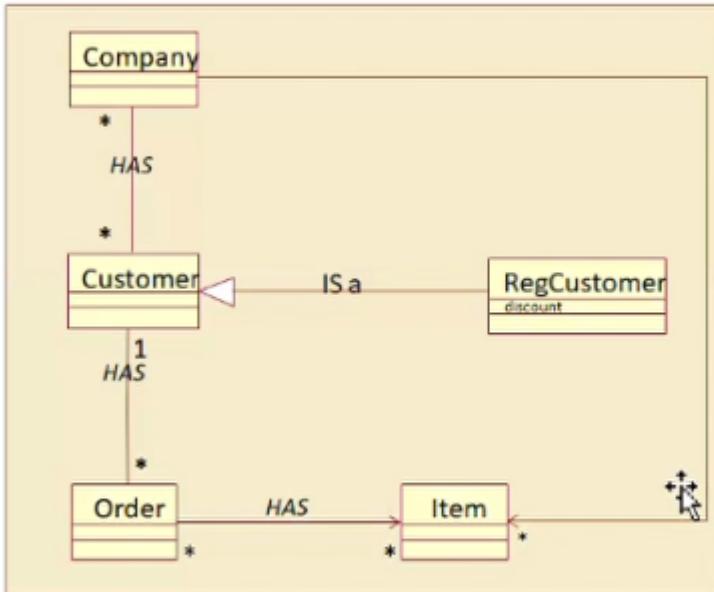


Identifying Relationships

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

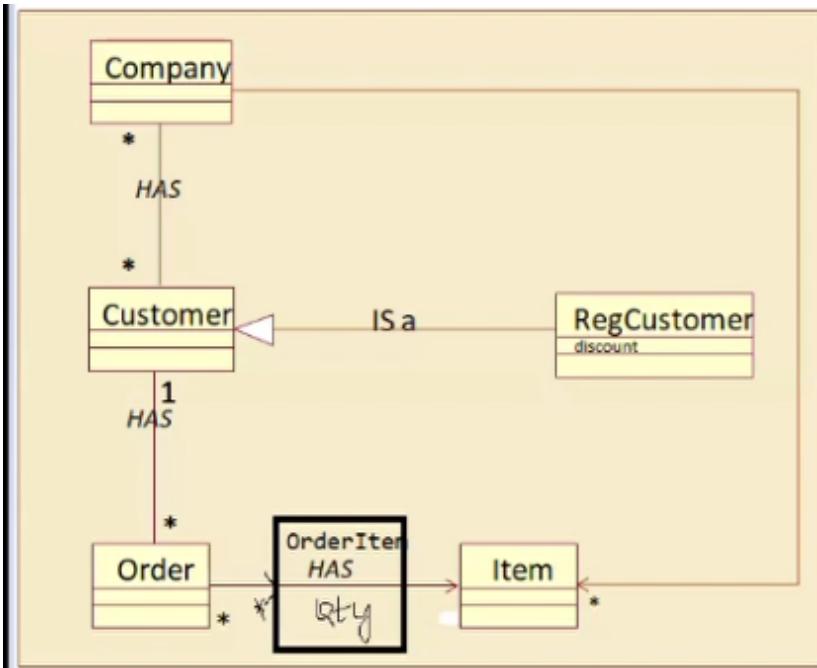


The OO Model



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



In the SkillAssure Assessment Framework,
Every course can have assessments
An Iteration has many courses and can also
have additional assessments
The training model has 4 Iterations
An assessment can be of multiple-choice
type,
hands-on exercise or project

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

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 *classification*!
 - *Classify* based on common behaviour or attributes
 - Agree on descriptions and names of useful *classes*
 - 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.



```
6 references
class Student
{
    public int I
    public string name;
}
```

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

```
6 references
class Student
{
    3 references
    public int id { get; set; }

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

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

The screenshot shows a code editor with three tabs at the top: "PassbyValueDemo2.cs", "ClassDemo.cs*", and "ConsoleApp*". The "ConsoleApp*" tab is active, showing the following C# code:

```
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace ConsoleApp
8   {
9       class ClassDemo
10      {
11          static void Main()
12          {
13              Student student = new Student();
14          }
15      }
16  }
17
```

A yellow vertical bar highlights the opening brace of the Main() method. A red box highlights the assignment statement "Student student = new Student();". A red double-headed arrow points from the text "Now these are two different statements" below to the highlighted assignment statement.

Now these are
two different statements

So i can just specify like this

```
ConsoleApp
```

```
2     using System.Collections.Generic;
3     using System.Linq;
4     using System.Text;
5     using System.Threading.Tasks;
6
7     namespace ConsoleApp
8     {
9         class ClassDemo
10        {
11            static void Main()
12            {
13                Student student; // so this is not creating a object this is just
14            }
15        }
16    }
17
```

**// so this is not creating a object this is just
a declaration**

```
assbyValueDemo2.cs ClassDemo.cs* ✘ ConsoleApp*  
ConsoleApp  
ConsoleApp.ClassDemo  
2     using System.Collections.Generic;  
3     using System.Linq;  
4     using System.Text;  
5     using System.Threading.Tasks;  
6  
7     namespace ConsoleApp  
8     {  
9         class ClassDemo  
10        {  
11            static void Main()  
12            {  
13                Student student; //declar  
14                student = new Student(); // ←  
15            }  
16        }  
17    }  
18}  
19
```

when you write this
this is where you are creating an instance

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

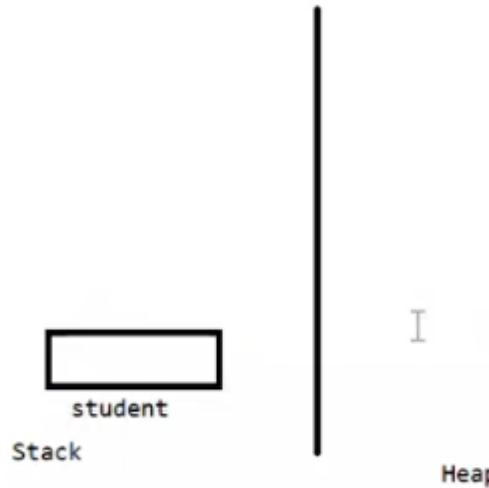
So internally the computers memory will have Stack and Heap

Stack

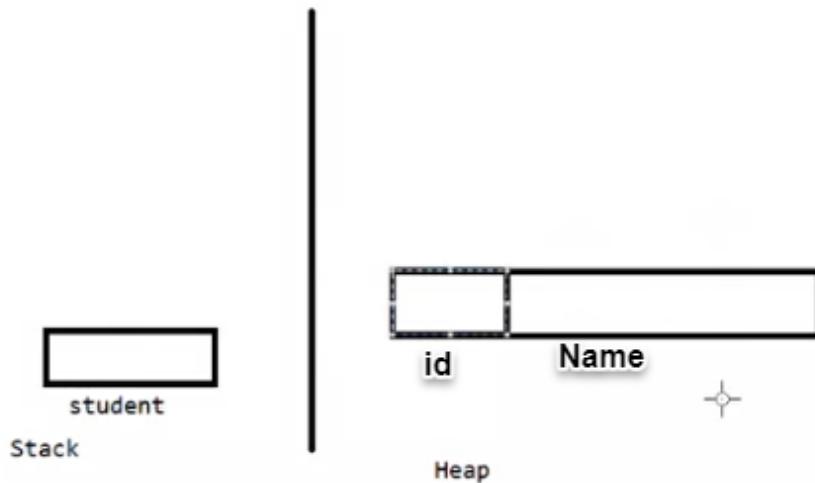
Heap

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**

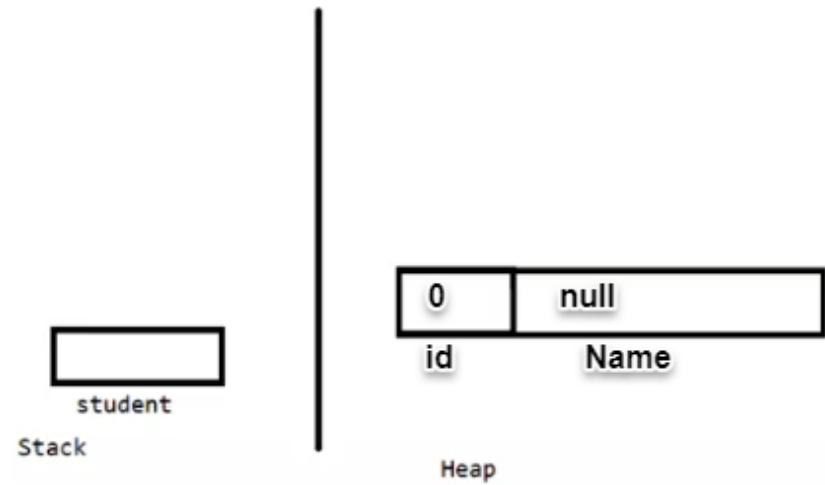
So id will be initialized to 0

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

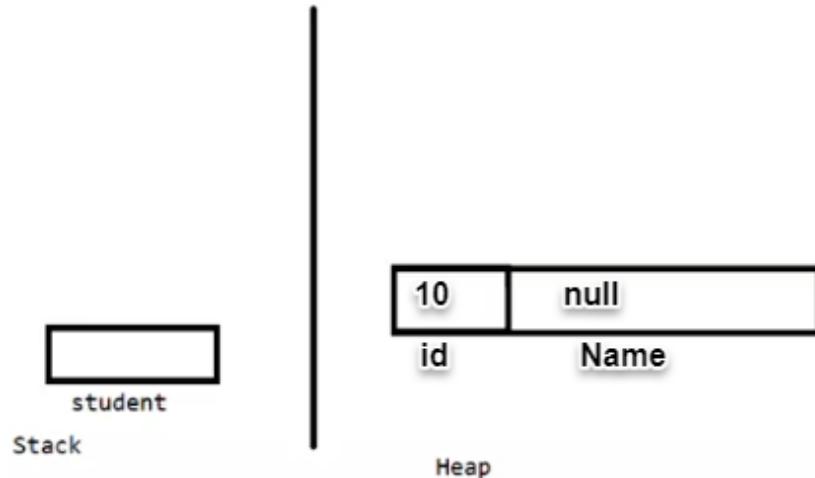


4. if you have explicit initialization that's get executed

```
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

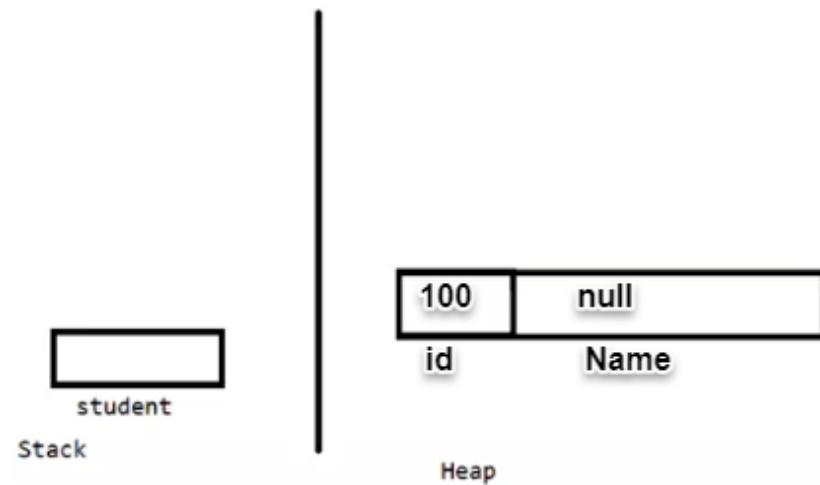


5. if you have a constructor that's get executed

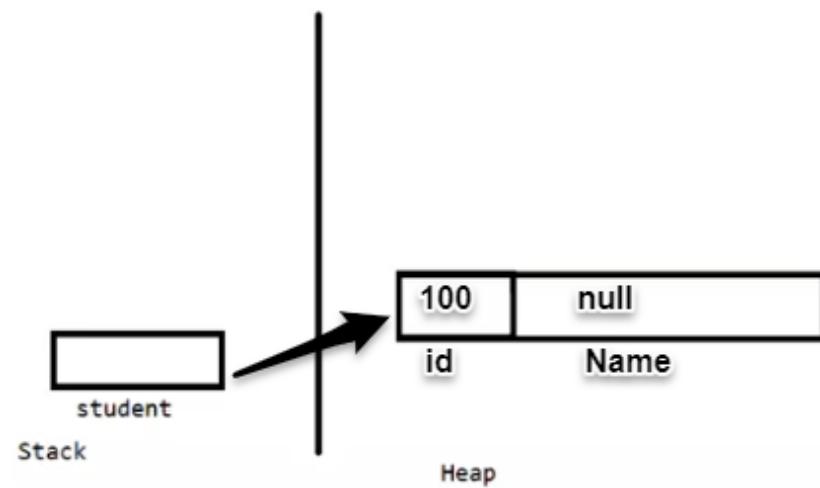
The screenshot shows the Visual Studio IDE interface with two open files: `ConsoleApp.cs` and `PassbyValueDemo2.cs`. The `ConsoleApp.cs` file is the active tab, displaying the following C# code:

```
7  namespace ConsoleApp
8  {
9      class Student
10     {
11         public int id { get; set; }
12         public string name { get; set; }
13
14         public Student()
15         {
16             id = 100;
17         }
18
19
20
21     class PassbyValueDemo2
22     {
23         static void Main()
24         {
25             Student student1 = new Student();
```

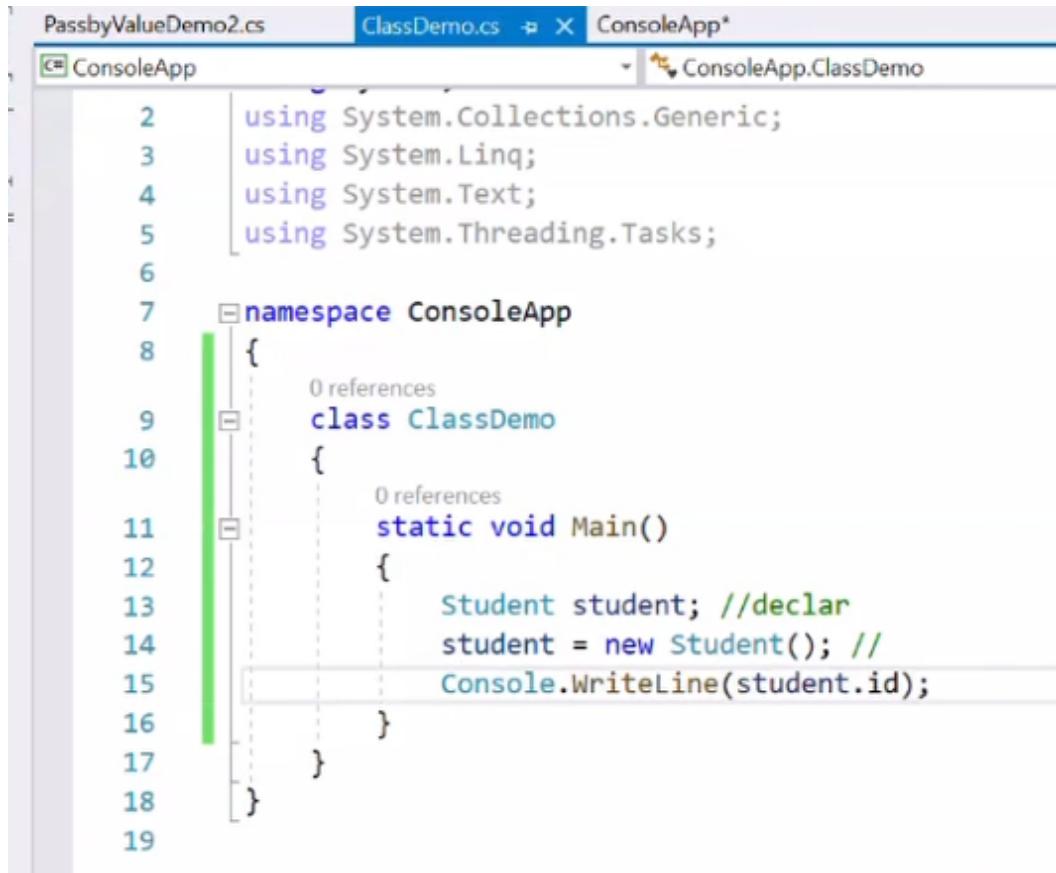
A red rectangular box highlights the constructor definition on lines 14-17. The code editor shows syntax highlighting and line numbers. The status bar at the bottom indicates "Item(s) Saved".



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



You can see this when i try to print this



The screenshot shows a code editor with three tabs at the top: "PassbyValueDemo2.cs", "ClassDemo.cs", and "ConsoleApp*". The "ConsoleApp*" tab is active, displaying the following C# code:

```
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace ConsoleApp
8  {
9      class ClassDemo
10     {
11         static void Main()
12         {
13             Student student; //declare
14             student = new Student(); //
15             Console.WriteLine(student.id);
16         }
17     }
18 }
19
```

A vertical green bar highlights the line "Console.WriteLine(student.id);". The code editor interface includes a status bar at the bottom.

so it print me 0 here

In java and .Net there is 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

```
7  namespace ConsoleApp
8  {
9      class Student
10     {
11         public int id { get; set; }
12         public string name { get; set; }
13
14         public Student()
15         {
16             id = 100;
17         }
18
19
20
21     class PassbyValueDemo2
22     {
23         static void Main()
24         {
25             Student student1 = new Student();
```

121 % No issues found

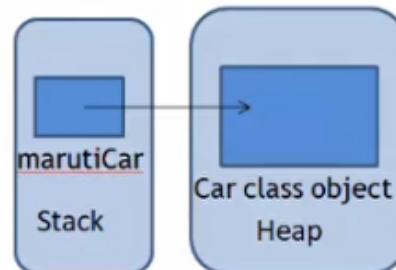
Output Package Manager Console Error List ... Immediate Window

Item(s) Saved

Type here to search

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



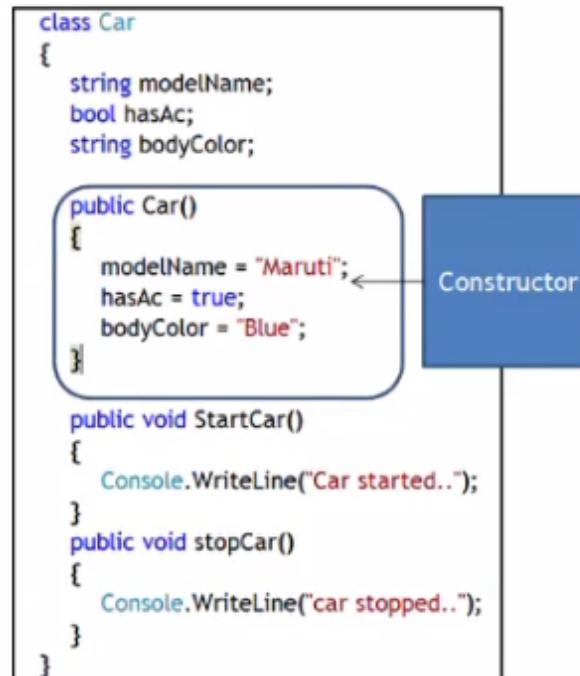
```
class Program
{
    static void Main()
    {
        Car marutiCar = new Car();
        marutiCar.StartCar();
        marutiCar.stopCar();

        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



```
L:\Users\yashwanth\Downloads\ConsoleApp1\src\ConsoleApp1\ClassDemo.cs
```

```
namespace ConsoleApp
{
    0 references
    class ClassDemo
    {
        0 references
        static void Main()
        {
            Student student; //declare
            student = new Student(); // ←
            Console.WriteLine(student.id);
        }
    }
}
```

So when you say new here
thats when the constructor gets
called

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

1. constructor should have same name as class name

```
/ references
class Student
{
    4 references
    public int id { get; set; }
    2 references
    public string name { get; set; }

    2 references
    public Student()
    {
    }

}
```

2. constructor should not have any return value

```
class Student
{
    4 references
    public int id { get; set; }
    2 references
    public string name { get; set; }
    0 references
    public void Student()
    {
        void
        [+] () ⌘V ⌘C ⌘O ⌘P ⌘G ⌘S ⌘L ⌘R ⌘F
    }
}
```

i cannot say void or intor any return type here

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

```
9 references
class Student
{
    4 references
    public int id { get; set; }
    2 references
    public string name { get; set; }

    2 references
    public Student() ← constructor with no parameter
    { }

    0 references
    public Student(int id) { } ← constructor with one parameter

    0 references
    public Student(int id, string name) { } ← constructor with 2 parameter
}
```

^{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

```
L:\Users\Asus\source\repos\ConsoleApp1\
```

```
namespace ConsoleApp
{
    0 references
    class ClassDemo
    {
        0 references
        static void Main()
        {
            Student student; //declare
            student = new Student(); //1
            Console.WriteLine(student.id);
        }
    }
}
```

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

The screenshot shows a Microsoft Visual Studio interface with the following details:

- Project:** ConsoleApp
- File:** PassbyValueDemo2.cs*
- Code Editor Content:**

```
7  namespace ConsoleApp
8  {
9      class Student
10     {
11         public int id { get; set; }
12         public string name { get; set; }
13
14         public Student(int id)
15         {
16             this.id = id;
17         }
18     }
19 }
```
- Toolbars and Menus:** Standard VS menus like File, Edit, View, Tools, etc.
- Toolbars:** Server Explorer, Toolbox, Task List, Solution Explorer, Properties, and Object Browser.
- Status Bar:** Shows the current file path: C:\Users\Public\Documents\Visual Studio 2019\Projects\ConsoleApp\PassbyValueDemo2.cs

PassbyValueDemo2.cs X ClassDemo.cs ConsoleApp

ConsoleApp

```
class Student
{
    public int id { get; set; }
    public string name { get; set; }

    public Student(int id)
    {
        this.id = id;
    }
}

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

Output Package Manager Console Error List Immediate Window

as soon as i do this i'll
get error here

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

A screenshot of the Microsoft Visual Studio IDE. The code editor window shows the following C# code:

```
Student student1 = new Student();
student1.id = 1001;
student1.name = "Shashi";
DispalyStudent(student1);
```

The word "Student" is underlined with a red squiggly line, indicating a potential error. A tooltip appears over "Student" with the text: "CS7036: There is no argument given that corresponds to the required formal parameter 'id' of 'Student.Student(int)'". Below the tooltip, there is a link "Show potential fixes (Alt+Enter or Ctrl+)." The status bar at the bottom of the IDE shows the time as "11:26 AM".

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

The screenshot shows a code editor window with the following code:

```
ConsoleApp.Student
class Student
{
    5 references
    public int id { get; set; }
    2 references
    public string name { get; set; }

    public Student() { }

    public Student(int id)
    {
        this.id = id;
    }
}

class PassbyValueDemo2
{
    static void Main()
    {
        Student student1 = new Student();
        student1.id = 1001;
    }
}
```

A red arrow points to the first constructor declaration, with the text "something like this" overlaid in red.

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

The screenshot shows the Visual Studio IDE with the 'ClassDemo.cs' file open. The code defines a class 'Student' with properties for 'id' and 'name', and a constructor that initializes 'id'. The code is color-coded: blue for keywords, green for comments, and yellow for the constructor name.

```
PassbyValueDemo2.cs*  X  ClassDemo.cs  ConsoleApp
ConsoleApp
ConsoleApp.Student
namespace ConsoleApp
{
    class Student
    {
        public int id { get; set; }
        public string name { get; set; }
        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);
        Student student2 = student1;
        student2.id = 1002;
        DispalyStudent(student2);
        DispalyStudent(student1);
    }
}

0 references
```

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

```
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace ConsoleApp
8  {
9      class ClassDemo
10     {
11         static void Main()
12         {
13             Student student; //declare
14             student = new Student(); //error
15             Console.WriteLine(student.id);
16         }
17     }
18 }
```

here also can you see
i'm getting error

```
App ConsoleApp.ClassDemo
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace ConsoleApp
8   {
9     class ClassDemo
10    {
11      static void Main()
12      {
13        Student student; //declare
14        student = new Student(1001); //here i'll pass 1001 here
15        Console.WriteLine(student.id); //here i'm printing id
16      }
17    }
18 }
```

```
U: C:\WINDOWS\system32\cmd.exe
US 1001
Press any key to continue . . .
na
```

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar includes tabs for 'ClassDemo.cs' and 'ConsoleApp'. The main editor window displays the following C# code:

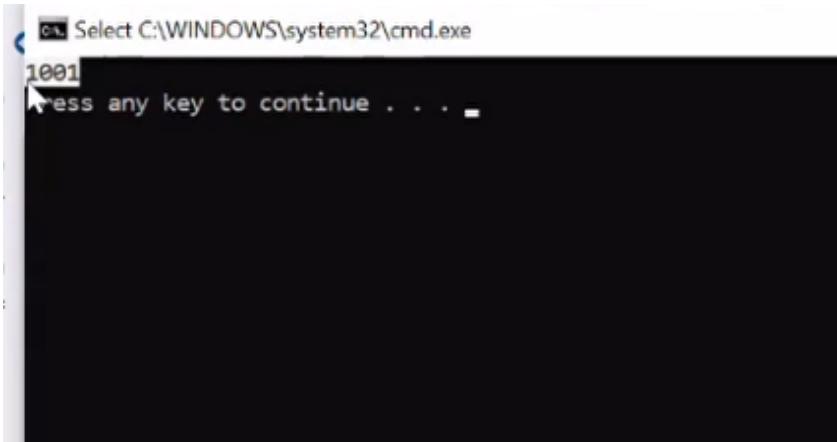
```
7  namespace ConsoleApp
8  {
9      class Student
10     {
11         public int id { get; set; } = 1; // Line 11
12         public string name { get; set; }
13         public Student(int id)
14         {
15             this.id = id;
16         }
17     }
18
19 
```

A red arrow points from the explanatory text 'if i do initialization here' to the assignment statement at line 11.

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 X ClassDemo.cs ConsoleApp

ConsoleApp

```
7  namespace ConsoleApp
8  {
9      class Student
10     {
11         public int id { get; set; }
12         public string name { get; set; }
13         public Student(int id)
14         {
15             this.id = id;
16         }
17
18         public Student(int id, string name)
19         {
20             this.id = id;           I
21             this.name = name;
22         }
23
24
25 }
```

0 references

121 % No issues found



```
PassbyValueDemo2.cs ClassDemo.cs* X ConsoleApp
ConsoleApp
ConsoleApp.ClassDemo
2     using System.Collections.Generic;
3     using System.Linq;
4     using System.Text;
5     using System.Threading.Tasks;
6
7     namespace ConsoleApp
8     {
9         0 references
10        class ClassDemo
11        {
12            0 references
13            static void Main()
14            {
15                Student student; //declare
16                student = new Student(1001); //
17                Console.WriteLine(student.id);
18                Console.WriteLine(student.name);
19            }
20        }
21    }
```

it will print me nothing

because i'm not initializing the particular name

so lets create a student

```
variedemo2.cs ClassDemos < X CONSUMAPP  
soleApp ConsoleApp.ClassDemo  
2     using System.Collections.Generic;  
3     using System.Linq;  
4     using System.Text;  
5     using System.Threading.Tasks;  
6  
7     namespace ConsoleApp  
8     {  
9         0 references  
10        class ClassDemo  
11        {  
12            0 references  
13            static void Main()  
14            {  
15                Student student; //declare  
16                student = new Student(1001); //  
17                Console.WriteLine(student.id);  
18                Console.WriteLine(student.name);  
19  
20                Student student1 = new Student(100, "Shashi");  
21                Console.WriteLine(student1.id);  
22                Console.WriteLine(student1.name);  
23            }  
        }  
    }  
No issues found
```

so now i'm passing
id as well as name

```
PS Select C:\WINDOWS\system32\cmd.exe
Pass 1001
100
Shashi
Press any key to continue . . .
```

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;
    }
}

Issues found | 🔍 ▾
  Error List  Immediate Window
```