Day 49 24/04/2025 .

Partial class

↳ two class with same name;

[These different parts all belong to the same class]

```
public class car {
}
```

-> Normal class.

```
public partial class car {
}
public partial class car {
}
```

-> Partial class.

* diffent class with same name.

↳ can only create one constructor as both class are same.

* both class will combined in compaile time.

Interface

*An interface is a blueprint that Contains only method signatures, properties, events or indexers, but no implementations.

* it like a contract that a class agrees to follow. it defines what a class should do.

→ product class : iproduct, icustomer

```
interface Iproduct {

    int get product () ;
    int getprize ();
    int getdiscount prize();
    string get name ();

}
```

after the method.

Object for product class

```
product prod = new product();
iproduct iprod = new product();
```

↳ only product method in iprod

```
icustomer icus = new product ();
```

after C# 12 Version we can give method definition in the interface

```
interface Icustomer
{
  int getcustomerid (int id);
  string getname (string name)

   {
        return name

    }

 }
```

---

abstract class

*similiar to interface.

*Cannot create object.

=> abstract class is like a combination of a normal class and an interface

* can only inherted.

```
abstract Class person
{
public
abstract int  countperson (int[] number);

public
abstract int  personid (int id);

public int Add (int a, int b)
{
        return a+b;

}

}
```

* we must definition the give method signatures to access the other method which are implemented.