ASSIGNMENT 10

1. Create an C# application as following:
   a. **Create an abstract class 'Employee' with specifications as followed:**

Private Fields:

Srno.	Identifiers	Data type
1	ID	String
2	BaseSalary	Int
3	WorkedDays	Int

Public fields:

Srno.	Identifiers	Data type
1	Fullname	String

Public Properties:

SrNo	Identifers	Type	Data type
1	pID	RW	String: "Exxxx" ,x: digit
2	pSalary	RW	>100 and <5000
3	pDays	RW	>0 and <=31

Public Abstract methods:

Srno.	Identifiers	Return type	Description
1	Display ( )	Void	Display detailed information
2	CalcSalary ( )	Int	Calculate actual salary

Public Methods

Srno.	Identifiers	Return type	Description
1	Input ( )	void	Input detailed information
2	ToString()	string	Override method ToString()

   b. **Create class 'Engineer' derived from 'Employee' :**
      i. Public Fields:

Srno.	Identifiers	Data type
1	Allowance	Int

      ii. Methods:
      Override methods in interface ICalc:
      Actual Salary = (Base Salary * WorkedDays)/24 + Allowance
      Override methods in base class:
      - Input(): revoke Input() of base class and after that, input value for allowance
      - Display() : print detailed information of an engineer

c. Create class 'EmployeeCatalog', implements named Iterator GetSenior.

i. Fields:

Srno.	Identifiers	Data type
1	eList	List <Engineer>

ii. Public Property:

Srno.	Identifiers	Type	Description
1	Add()	W	Add a new engineer into list (eList)

iii. Public Methods:

Srno.	Identifiers	Return	Description
2	DisplayAll( )	void	Return list of èngineer by appropriate format
3	GetSenior ( )	IEnumerable	Return list of èngineers having actual salary >= 500

d. Create menu-based client class Test for testing class EmployeeCatalog.