AP - ASSIGNMENT 8

1. Create an C# application as following:

   a. Define a delegate named "DValid" :

Identifiers	*Return Type*	*Parameter*
DValid	void	String

   b. Create an abstract class 'Employee' with specifications as followed:

Event

Identifiers	*Type*	*Description*
eValid	DValid	Raise when invalid data entered into the properties.

Private **Fields:**

Srno.	Identifiers	Data type
1	ID	string
2	Fullname	string
3	BaseSalary	int
4	WorkedDays	Int

Public **Properties:**

SrNo	Identifers	Type	Data type, validate
1	pID	RW	String: "Exxxx" ,x: digit
2	pName	RW	Not null
3	pSalary	RW	Int, >100 and <5000
4	pDays	RW	Int, >0 and <=31

Public Abstract **Methods**:

Srno.	Identifiers	Return type	Description
1	Display ( )	void	Display detailed information

Public **Methods**

Srno.	Identifiers	Return type	Description
1	Input ( )	void	Input detailed information
2	ToString()	string	Override method ToString()
3	Validate(String s)	void	Handle event eValid by throw an exception with message s

   c. Create an interface 'ICalc' :

Srno.	Identifiers	Return type	Description
1	CalcSalary ( )	int	Calculate actual salary

d. **Create class ‘Engineer’ derived from ‘Employee’, implements interface ‘ICalc’:**

   i. Public **Fields:**

Srno.	Identifiers	Data type
1	Allowance	Int

   ii. Methods:

- Override method **CalcSalary ( )** of 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

e. **Create class ‘EmployeeList’, implements a generic Iterator.**

   Event

Identifiers	*Type*	*Description*
eEmpty	DValid	Raise when list of engineers is empty.

   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:

No.	Identifiers	Return	Description
1	Remove(String id)	void	Return list of engineer by appropriate format
2	GetEnumerator ( )	IEnumerator<Engineer>	Return all of engineers
3	EmptyList(String s)	void	Handle event eEmpty by throw an exception with message s

f. **Create menu-based client class Program for testing class EmployeeList as following:**

   1. **Add New Engineer**
   2. **Remove An Engineer by ID**
   3. **Display All Engineers**
   4. **Display All Senior Engineers (basic salary > 2000)**
   5. **Exit**