.NET ASSIGNMENT -03

Prajwal_77

Q1 Create the following classes

Employee

Prop

string Name -> no blanks

int EmpNo -> readonly, autogenerated

short DeptNo -> > 0

abstract decimal Basic

Methods

abstract decimal CalcNetSalary()

Manager : Employee

Prop

string Designation -> cant be blank

GeneralManager : Manager

Prop

string Perks -> no validations

CEO : Employee

Make CalNetSalary() a sealed method

NOTE : Overloaded constructors in all classes calling their base class constructor

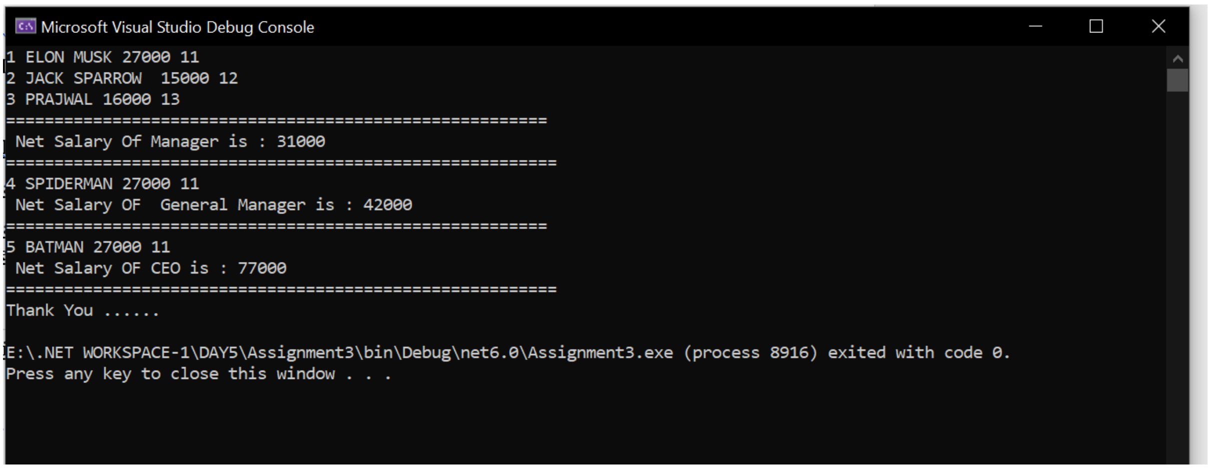
All classes must implement IDbFunctions interface

All classes to override the abstract members defined in the base class(Employee). Basic property to have different validation in different classes.

Solution –

```
namespace Employee3
{
    internal class Program
    {
        static void Main(string[] args)
        {

            Manager manager1 = new Manager("Manager1", "ELON MUSK", 27000, 11);
            Manager manager2 = new Manager("Manager2", "JACK SPARROW ", 15000,
12);
            Manager manager3 = new Manager("Manager3", "PRAJWAL", 16000, 13);


Console.WriteLine("======================================================== ");

            decimal netSal = manager3.CalculateNetSalary();
            Console.WriteLine(" Net Salary Of Manager is : " + netSal);


Console.WriteLine("========================================================= ");

            Manager generalManager = new GeneralManager("Holiday_package",
"GManager", "SPIDERMAN", 27000, 11);
            netSal = generalManager.CalculateNetSalary();
            Console.WriteLine(" Net Salary OF  General Manager is : " + netSal);


Console.WriteLine("======================================================== ");

            CEO ceo = new CEO("BATMAN", 27000, 11);
            netSal = ceo.CalculateNetSalary();
            Console.WriteLine(" Net Salary OF CEO is : " + netSal);


Console.WriteLine("========================================================= ");

            Console.WriteLine("Thank You ......");

        }

        interface IDbFunctions
        {
            void create();

        }
        internal abstract class Employee3
        {
            private int empNo;

            public int EmpNo
            {
                set
                {
                    if (value > 0)
                        empNo = value;
                    else
                        Console.WriteLine("invalid emp no");
                }
                get
                {
                    return empNo;
                }
            }


            private string name;
            public string Name
            {
                set
                {
                    if (value != null)
                    {
                        name = value;
                    }
                    else
                    {
                        Console.WriteLine("blank names r not allowed");
                    }

                }
                get
                {
                    return name;
                }

            }

            public abstract decimal Basic
            {
                set;
                get;
            }


            private short deptNo;
            public short DeptNo
            {
                set
                {
                    if (value > 0)
                        deptNo = value;
                    else
                        Console.WriteLine("invalid Dept No");
                }
                get
                {
                    return deptNo;
                }
            }

            private static int nextEmpNo = 1;


            public abstract decimal CalculateNetSalary();

            public Employee3(string Name = "raj", decimal Basic = 10000, short
DeptNo = 10)
            {

                this.empNo = nextEmpNo;
                nextEmpNo++;
                this.Name = Name;
                this.Basic = Basic;
                this.DeptNo = DeptNo;

                Console.WriteLine(empNo + " " + this.Name + " " + this.Basic + "
" + this.DeptNo);
            }


        }

        internal class Manager : Employee3, IDbFunctions
        {
            private string designation;
            public string Designation
            {
                set
                {
                    if (value != null)
                    {
                        designation = value;
                    }
                    else
                    {
                        Console.WriteLine("blank names r not allowed");
                    }

                }
                get
                {
                    return designation;
                }

            }

            private decimal basic;
            public override decimal Basic
            {
                set
                {
                    if (value >= 10000 && value <= 100000)

                        basic = value;
                    else
                        Console.WriteLine("invalid basic");
                }
                get
                {
                    return basic;
                }

            }


            public Manager(string Designation, string Name, decimal Basic, short
DeptNo) : base(Name, Basic, DeptNo)
            {

                this.Designation = Designation;
                this.Basic = Basic;

            }

            public override decimal CalculateNetSalary()
            {
                int allowances = 30000;
                int deductions = 15000;



                decimal grossSalary = basic + allowances;

                decimal netSal = grossSalary - deductions;

                return netSal;
            }

            public void create()
            {
                throw new NotImplementedException();
            }
        }

        internal class GeneralManager : Manager, IDbFunctions
        {
            public string Perks;

            public GeneralManager(string Perks, string Designation, string Name,
decimal Basic, short DeptNo) : base(Designation, Name, Basic, DeptNo)
            {
                this.Perks = Perks;
            }
        }

        internal class CEO : Employee3, IDbFunctions
        {
            private decimal basic;
            public override decimal Basic
            {
                set
                {
                    if (value >= 10000 && value <= 150000)

                        basic = value;
                    else
                        Console.WriteLine("invalid basic");
                }
                get
                {
                    return basic;
                }

            }

            public CEO(string Name, decimal Basic, short DeptNo) : base(Name,
Basic, DeptNo)
            {
                this.Basic = Basic;

            }

            public sealed override decimal CalculateNetSalary()
            {
                int allowances = 70000;
                int deductions = 20000;

                decimal grossSalary = basic + allowances;
                decimal netSal = grossSalary - deductions;

                return netSal;
            }

            public void create()
            {
                throw new NotImplementedException();
            }
        }



    }
}
```

Output-



```
1 ELON MUSK 27000 11
2 JACK SPARROW  15000 12
3 PRAJWAL 16000 13
=======================================================
 Net Salary Of Manager is : 31000
========================================================
4 SPIDERMAN 27000 11
 Net Salary OF  General Manager is : 42000
=======================================================
5 BATMAN 27000 11
 Net Salary OF CEO is : 77000
========================================================
Thank You ......

E:\.NET WORKSPACE-1\DAY5\Assignment3\bin\Debug\net6.0\Assignment3.exe (process 8916) exited with code 0.
Press any key to close this window . . .
```