Day 10 Evening Assignment

By Manoj Karnatapu - NBHealthCareTechnologies

Assignment 1
Write the 2 main uses of Abstract class by using the example discussed in the class.
Answer

- ✓ Abstract class is used for Code-Reusability.
- ✓ What ever we have normal methods in the abstract class, we can use them with ease in the derived class. Where as , the abstract methods in the abstract class must be implemented & overridden in the derived class mandatory.

OR

- ✓ Enforcing the derived class, must override the abstract methods.

Assignment 2
Write C# Code, for demonstrating abstract class.
Code

```csharp
using System;

// Author : Manoj.Karnatapu
// Purpose : Abstract Template Creation

// For Reference, check Day10EVEProject1 in the same Repository.

namespace Day10EVEProject1
{
    /// <summary>
    /// This is an Abstract Class
    /// </summary>
    abstract class Salary
    {
        /// <summary>
        /// This is a GetPF Normal Method with Body
        /// </summary>
        /// <param name="basic">INT Basic</param>
        /// <returns>Cal O/p</returns>
        public int GetPF(int basic)
```

```csharp
        { return 12 * basic / 100; }
        /// <summary>
        /// This is a GetHRA Normal Method with Body
        /// </summary>
        /// <param name="basic">INT Baisc</param>
        /// <returns>Cal O/p</returns>
        public int GetHRA(int basic)
        { return 40 * basic / 100; }

        /// <summary>
        /// This is an Abstract Method GetCA, it doesn't have a Body in
Abstract class.
        /// </summary>
        /// <returns>Enforcing Derived class to Use this method
mandatorly.</returns>
        public abstract int GetCA();
        /// <summary>
        /// This is an Abstract Method GetSA, it doesn't have a Body in
Abstract class.
        /// </summary>
        /// <returns>Enforcing Derived class to Use this method
mandatorly.</returns>
        public abstract int GetSA();
    }

    class Microsoft : Salary
    {
        public override int GetCA()
        {
            return 6000;
        }

        public override int GetSA()
        {
            return 7000;
        }
    }
    class Google : Salary
    {
        public override int GetCA()
        {
            return 10000;
        }

        public override int GetSA()
        {
            return 10000;
        }
    }
    class IBM : Salary
    {
        public override int GetCA()
        {
            return 4000;
        }

        public override int GetSA()
        {
            return 6000;
        }
    }
    class Facebook : Salary
    {
```

```csharp
        public override int GetCA()
        {
            return 20000;
        }

        public override int GetSA()
        {
            return 20000;
        }
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            // Microsoft Code

            // Google Code

            // IBM Code

            // Facebook Code

            Console.WriteLine("Abstract Class Template Execution Success");

            Console.ReadLine();
        }
    }
}
```

Output