Day10 Evening Assignment

By
Narala Praveen

04-02-2022

Question 1:
Research and try to understand what is Abstraction?

a. An Abstract Base class cannot be instantiated(It means the object of that class cannot be created).
b. Class having the Abstract keyword with some of its methods (not all)is known as an abstract base class.
c. Class having the abstract keyword with all of its methods is known as pure base abstract base class.
d. The method of the abstract class that has no implementation is known as ”operation” it can be defined as an abstract void method();.
e. An abstract class holds the methods but the actual implementation of those methods is made in derived class.

Question 2:
Write the 2 main uses of abstract class by using the example discussed in the class?

Uses:

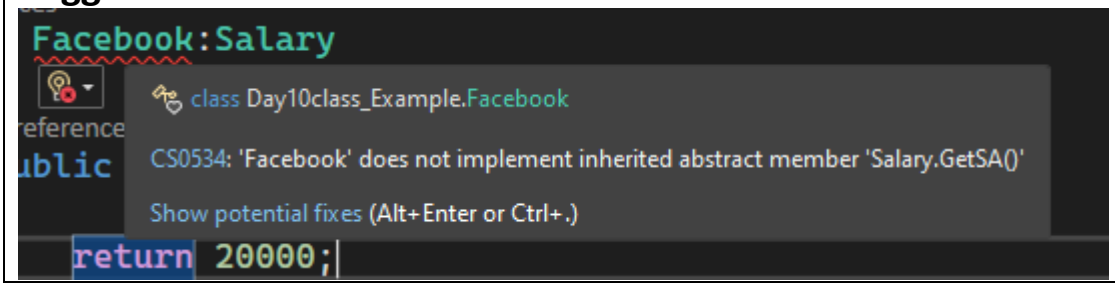
a. Reusability of code.
b. Enforcing the variables of abstract base class to the derived class(An abstract class as a template when we forget to implement the methods of abstract base class you will get error)

Example code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day10class_Example
{    //*****************************************************************\\
    //Author:Narala Praveen
    //Purpose:To Create an Abstract class.
    //*****************************************************************\\

    abstract class Salary
    {/// <summary>
    /// This method is for getting PF
    /// </summary>
    /// <param name="basic"></param>
    /// <returns></returns>
        public int GetPF(int basic)
        {
            return 12 * basic / 100;
        }
        /// <summary>
        /// this method is to get HRA
        /// </summary>
        /// <param name="basic"></param>
        /// <returns></returns>
        public int GetHRA(int basic)
        {
            return 40 * basic / 100;
        }
        /// <summary>
        /// This method to override Convience allowance in derived class
        /// </summary>
        /// <returns></returns>
        public abstract int GetCA();

        /// <summary>
        /// This method is to override Special allowance in derived class
        /// </summary>
        /// <returns></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 100000;
        }
```

```
        public override int GetSA()
        {
            return 10000;
        }
    }
    class IBM:Salary
    {
        public override int GetCA()
        {
            return 4000;
        }
        public override int GetSA()
        {
            return 6000;
        }
    }
    class Facebook:Salary
    {
        public override int GetCA()
        {
            return 20000;
        }
        public override int GetSA()
        {
            return 8000;
        }
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            //MicroSoft
            //Google
            //IBM
            //Facebook
            Console.WriteLine("Comleted Processing");
            Console.ReadLine();
        }
    }
}
```

Suggestion:



Question3:
Create an Example of your own choice to demonstrate abstract class?

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day10Abstract_Own_Example
{    //*****************************************************************\\
    //Author:Narala Praveen
    //Purpose:To Create an Abstract class of own choice
    //*****************************************************************\\

    abstract class Car
    {/// <summary>
     /// This method is for getting car metirial
     /// </summary>
     /// <param name="basic"></param>
     /// <returns></returns>
        public void  PrintMaterial(string iron)
        {
            Console.WriteLine("castiron");
        }
        /// <summary>
        /// this method is to get number of wheels
        /// </summary>
        /// <param name="basic"></param>
        /// <returns></returns>
        public int GetWheels(int n)
        {
            return 4 ;
        }
        /// <summary>
        /// This method to override model in derived class
        /// </summary>
        /// <returns></returns>
        public abstract void  GetModel();

        /// <summary>
        /// This method is to override price in derived class
        /// </summary>
        /// <returns></returns>
        public abstract int GetPrice();

    }
    class BMW : Car
    {
        public override void GetModel()
        {
            Console.WriteLine("BMWX3");
        }
        public override int GetPrice()
        {
            return 6500000;
        }
    }
    class Audi: Car
    {
```

```
        public override void GetModel()
        {
            Console.WriteLine("AudiQ7");
        }
        public override int GetPrice()
        {
            return 7900000;
        }
    }
    class Mercedes : Car
    {
        public override void GetModel()
        {
            Console.WriteLine("Limousine");
        }
        public override int GetPrice()
        {
            return 5000000;
        }
    }
    class Toyota : Car
    {
        public override void GetModel()
        {
            Console.WriteLine("Vellfire");
        }
        public override int GetPrice()
        {
            return 8000000;
        }



    }
    internal class Program
    {
        static void Main(string[] args)
        {
            //BMW
            //Audi
            //Mercedes
            //Toyota
            Console.WriteLine("Comleted Processing");
            Console.ReadLine();
        }
    }
}
```