Day 10 Evening Assignment
(4th Feb 2022)

by
RamCharan

1.Research and try to understand what is Abstraction.

- Abstraction is process of hiding unnecessary data and showing only necessary data.
- It reduces code complexity.
- It is one of the main important aspect in the Object Oriented Programming.
- We can implement Abstraction using abstract classes and interfaces.
- Keyword for implementing Abstraction is “ abstract ”.

2.Write 2 main uses of Abstarct class by using example.

- Reusability

- Enforcing the derived class to must override abstract methods in the base class.

Example: Demonstrating abstract class.

Code:

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

namespace Day10Project6
{
    //Author : Rc
    /*****Purpose: Abstraction******/
    /// <summary>
    /// Abstract Parent class
    /// </summary>
    abstract class Salary
    {
        /// <summary>
        /// This method is to get PF of the employee
    /// </summary>
    /// <param name="basic"></param>
    /// <returns>PF</returns>
    public int GetPf(int basic)
    {
        return 12*basic/100;
    }
    /// <summary>
    /// This method is to get HouseRentalAllowances of the employee
    /// </summary>
    /// <param name="basic"></param>
    /// <returns></returns>
    public int GetHRA(int basic)
    {
        return 40*basic/100;
    }
    /// <summary>
    /// This abstract method is to get ConvenienceAllowances of the employee
    /// </summary>
    /// <returns></returns>
    public abstract int GetCA();
    /// <summary>
    /// This abstract method is to get SpecialAllowances of the employee
    /// </summary>
    /// <returns></returns>
    public abstract int GetSA();
}
class Google : Salary
{
    public  override int GetCA()
    {
        return 4000;
    }
    public override int GetSA()
    {
        return 6000;
    }
}
class Microsoft : Salary
{
    public override int GetCA()
    {
        return 8000;
    }
    public override int GetSA()
    {
        return 4000;
    }
}
class IBM : Salary
{
    public override int GetCA()
```

```
        {
            return 10000;
        }
        public override int GetSA()
        {
            return 10000;
        }
    }
    class Facebok : Salary
    {
        public override int GetCA()
        {
            return 2000;
        }
        public override int GetSA()
        {
            return 3000;
        }
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            //Google


            //MicroSoft


            //IBM


            //Facebook

            Console.WriteLine("Completed Processing");

            Console.ReadLine();

        }
    }
}
```

3.Create example of our choice and demonstrate abstract class.

Code:

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

namespace Day10Project7
{
    //Author : Rc
    /********Purpose: Demonstrating abstract class with Vehicles*********/
    //Abstract Parent class
    abstract class Vehicle
    {
        /// <summary>
        /// This method is to say usage
        /// </summary>
        public void Usage()
        {
            Console.WriteLine("Used to Trasport");
        }
        /// <summary>
        /// This method is for suggesting to carry License
        /// </summary>
        public void License()
        {
            Console.WriteLine("Please always carry your License");
        }
        /// <summary>
        /// This method is to print numbers of wheels in a given vehicle
        /// </summary>
        /// <returns>Count of wheels</returns>
        public abstract  int wheels();
    }
    class Bike : Vehicle
    {
        public override int wheels()
        {
            return 2;
        }
    }
    class Auto : Vehicle
    {
        public override int wheels()
        {
            return 3;
        }
    }
    class Car : Vehicle
    {
        public override int wheels()
        {
            return 4;
        }
    }
    class Bus : Vehicle
    {
        public override int wheels()
        {
            return 6;
        }
    }

    internal class Program
    {
        static void Main(string[] args)
        {
            //Bike


            //Auto


            //Car


            //Bus


            Console.WriteLine("Happy Journey:");


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

END OF THE DAY