NB Healthcare Technologies Pvt Ltd

Day 10 Evening Assignment (4 – Feb- 2022)
By
Vamsi Krishna Mandapati

1. Research and try to understand what is Abstraction

Abstraction :

abstraction is the process of hiding certain details and showing only essential information to the user.
Abstraction can be achieved with either **abstract classes** .

The `abstract` keyword is used for classes and methods:

- **Abstract class:** is a restricted class that cannot be used to create objects (to access it, it must be inherited from another class).

- **Abstract method:** can only be used in an abstract class, and it does not have a body. The body is provided by the derived class (inherited from).

To achieve security - hide certain details and only show the important details of an object.

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

Two Main Uses of Abstract Class:
1. Code Reusability: in the below example , in salary class we mentioned two methods which are GetPF() and GetHRA(), these two methods we write once in base class and used in derived classes.
2. Enforcing the Derived Class ,to must override the abstract methods.

Code:

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

namespace Day10Project1
{
    abstract class Salary
    {
        public int GetPF(int basic)
        {
            return 12 * basic / 100;
        }

        public int GetHRA(int basic)
        {
            return 40 * basic / 100;
        }

        public abstract int GetCA();
        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
    {


        public override int GetCA()
        {
            return 20000;
        }


    }

    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Completed Processing");
        }
    }
}
```

3. Create one more example of your choice to demonstrate abstract class

Code:

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

namespace Day10Project1
{
    abstract class EmployeeDetails
    {
        public string Name(string name)
        {
            return name;
        }
        public int Id(int id)
        {
            return id;
        }

        public abstract string Mobile();
        public abstract string Email();

    }

    class Microsoft : EmployeeDetails
    {


        public override string Mobile()
        {
            return "854534848";
        }

        public override string Email()
        {
            return "abc@gmail.com";

        }
    }

    class Google : EmployeeDetails
    {


        public override string  Mobile()
        {
            return "344684889";
        }

        public override string Email()
        {
            return "123@gmail.com";
        }

    }

    class IBM : EmployeeDetails
    {


        public override string Mobile()
        {
            return "5425468464";
        }

        public override string Email()
        {
            return "sree@gmail.com";
        }
    }

    class Facebook : EmployeeDetails
    {


        public override string Mobile()
        {
            return "45584447585";
        }


    }

    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Completed Processing");
        }
    }
}
```