DAY10 EVENING ASSIGNMENT
BY
PAVAN KUMAR (04-02-2022)

Q1). Research and try to understand what is Abstraction

ABSTRACTION: It is used to hide the implementation details. Abstract classes are base classes with partial implementation.

- Enforcing the derived class, must override the abstract methods.
- We cannot declare an abstract method outside an abstract class.
- We cannot create an instance of an abstract class.

PROJECT: 1

WRITE A PROGRAM USING ABSTRACT CLASS

CODE:

```
using System;
using System.Collections;

// DONE BY: PAVAN
// Purpose:  Creation OF Abstract Template


namespace Day10EVEProject1
{
   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)
      {return 12 * basic / 100;}
      /// <summary>
      /// This is a Get HRA Normal Method with Body
      /// </summary>
      /// <param name="basic">INT Basic</param>
      /// <returns>Cal O/p</returns>
      public int GetHRA(int basic)
      {
         return 40 * basic / 100;
      }

      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 mandatorily.</returns>
    public abstract int GetSA();
}

class Microsoft: Salary
{

    public override int GetCA()
    {
        return 9000;
    }

    public override int GetSA()
    {
        return 6000;
    }
}
class Google: Salary
{
    public override int GetCA()
    {
        return 8000;
    }

    public override int GetSA()
    {
        return 5000;
    }
}
class IBM: Salary
{
    public override int GetCA()
    {
        return 4000;
    }

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

```
        return 5000;
      }

      public override int GetSA()
      {
        return 1000;
      }
    }
    internal class Program
    {
      static void Main(string[] args)
      {


        Console.WriteLine("TRANSACTION SUCCESSFUL");

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

OUTPUT: