Lab Assignment

Program 1-

```csharp
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Assignment2

{

    internal class Program

    {

        static void Main(string[] args)

        {

            Employee e1 = new Employee(1, "Ritik", 123.56);

            e1.displayData();

        }

    }

    class Employee

    {

        public int id;

        public string name;

        public double salary;


        public Employee(int id, string name, double salary)

        {

            this.id = id;
```

```csharp
      this.name = name;

      this.salary = salary;

   }

   public void displayData()

   {

      Console.WriteLine("id :" + this.id);

      Console.WriteLine("Name :" + this.name);

      Console.WriteLine("Salary :" + this.salary);

      Console.ReadKey();

   }

 }

}
```

Program 2 -

```csharp
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Assignment2

{

   internal class Program

   {

      static void Main(string[] args)

      {

         Bank b1 = new Bank(12345, "Ritik", 1000);

         b1.displayData();

      }

   }
```

```csharp
class Bank
{
    public int AccountNumber;

    public string name;

    public double balance;


    public Bank(int AccountNumber, string name, double balance)
    {
        this.AccountNumber = AccountNumber;

        this.name = name;

        this.balance = balance;
    }
    public void deposit(double balance)
    {
        balance += balance;
    }
    public void withdrawl(double amount)
    {
        if (amount > balance)
        {
            Console.WriteLine("Insufficinent Balance");
        }
        else
        {
            balance = balance - amount;
        }
    }
    public void displayData()
    {
        Console.WriteLine("AccountNumber:" + this.AccountNumber);

        Console.WriteLine("Name :" + this.name);
```

```csharp
            Console.WriteLine("Balance :" + this.balance);

            Console.ReadKey();

        }

    }

}
```

Program 3-

```csharp
using System;

using System.Collections.Generic;

using System.Diagnostics.CodeAnalysis;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Assignment2

{

    internal class Program

    {

        static void Main(string[] args)

        {

            float[] nums = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

            MathHelper.average(nums);

        }

    }


    public static class MathHelper

    {

        static float sum= 0;

        static float avg = 0;

        public static void average(float[] nums)
```

```csharp
        {
            for(int i=0;i<nums.Length; i++)
            {
                sum += nums[i];
            }
            avg = sum/nums.Length;
            Console.WriteLine("Average is " + avg);
            Console.ReadKey();
        }

    }

}
```

Program 4 -

```csharp
using System;
using System.Collections.Generic;
using System.Diagnostics.CodeAnalysis;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading.Tasks;

namespace Assignment2
{
    internal class Program
    {
        static void Main(string[] args)
        {
```

```csharp
            Logger.setNameAndPassword("Ritik", 1234);

            Logger.Login("Ritik", 1234);


        }
    }


    public static class Logger
    {
        static string name;
        static int password;
        public static void setNameAndPassword(string username ,int userpassword)
        {
            name = username;
            password = userpassword;
        }
        public static void Login(string username ,int userpassword)
        {
            if (username == name && userpassword == password)
            {
                Console.WriteLine("Login SuccessFull");
                Console.ReadKey();
            }
            else
            {
                Console.WriteLine("Login Failed");
                Console.ReadKey();
            }
        }
    }

}
```

Program 5-

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

namespace Assignment2
{
    public partial class Person
    {
        public string firstName;
        public string lastName;
        public Person(string firstName, string lastName)
        {
            this.firstName = firstName;
            this.lastName = lastName;
        }

    }
}

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

namespace Assignment2
```

```csharp
{
    public partial class Person
    {
        public void showDetails() {

            Console.WriteLine("FirstName :" +firstName+ "LastName :" + lastName);

            Console.ReadKey();

        }
    }
}



using System;

using System.Collections.Generic;

using System.Diagnostics.CodeAnalysis;

using System.Linq;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading.Tasks;


namespace Assignment2
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Person p1 = new Person("Ritik", "Kaushik");
            p1.showDetails();


        }
```

```
        }

    }


Program 6 -

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Assignment2
{
    public partial class Employee
    {
        public int id;
        public string name;
        public double salary;

        public Employee(int id, string name, double salary)
        {
            this.id = id;
            this.name = name;
            this.salary = salary;
        }

    }
}


using System;
```

```csharp
using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Assignment2

{

    public partial class Employee

    {


        public void displayData()

        {

            Console.WriteLine("id :" + this.id);

            Console.WriteLine("Name :" + this.name);

            Console.WriteLine("Salary :" + this.salary);

            Console.ReadKey();

        }

    }

}


using System;

using System.Collections.Generic;

using System.Diagnostics.CodeAnalysis;

using System.Linq;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading.Tasks;


namespace Assignment2

{

    internal class Program
```

```
    {
        static void Main(string[] args)

        {
            Employee e1 = new Employee(1, "Ritik", 1000);

            e1.displayData();

        }


    }

}
```

Program 7 -

```
using System;

using System.Collections.Generic;

using System.Diagnostics.CodeAnalysis;

using System.Linq;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading.Tasks;


namespace Assignment2

{
    internal class Program

    {
        static void Main(string[] args)

        {
            Circle c1 = new Circle();

            c1.setRadius(5);

            c1.getarea();

        }
```

```csharp
    }

    public abstract class Shape
    {
        public abstract void getarea();
    }
    public class Circle : Shape
    {
        float radius;
        public void setRadius( float r)
        {
            radius = r;
        }
        public override void getarea()
        {
            double area = 3.14 * radius * radius;
            Console.WriteLine("Area is :" + area);
            Console.ReadKey();
        }
    }
}
```

Program 8 -

```csharp
using System;
using System.Collections.Generic;
using System.Diagnostics.CodeAnalysis;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading.Tasks;
```

```csharp
namespace Assignment2
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Dog d1 = new Dog();
            Cat c1 = new Cat();
            d1.Sound("WOW WOW");
            c1.Sound("MEOW MEOW");
        }

    }

    public abstract class Animal
    {
        public abstract void Sound(string sound);
    }
    public class Dog : Animal
    {
        public override void Sound(string sound)
        {
            Console.WriteLine("Dog is barking " + sound);
            Console.ReadKey();
        }
    }
    public class Cat: Animal
    {
        public override void Sound(string sound)
        {
```

```
            Console.WriteLine("Cat Sound " + sound);

            Console.ReadKey();

        }


    }



}
```

Program 9 -

```
using System;

using System.Collections.Generic;

using System.Diagnostics.CodeAnalysis;

using System.Linq;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading.Tasks;


namespace Assignment2

{

    internal class Program

    {

        static void Main(string[] args)

        {

            Car Audi = new Car("Audi");
```

```csharp
        }

    }

    public sealed class Vechile
    {
        string vechile;
        public void startEngine(string vechile)
        {
            this.vechile = vechile;
            Console.WriteLine(vechile + " is start");
        }
        public void stopEngine()
        {


            Console.WriteLine(vechile + "is stop");
        }
    }

    public class Car : Vechile
    {
        string name;
        public Car(string name)
        {
            this.name = name;
        }
    }

}
```

Program 10-

```csharp
using System;

using System.Collections.Generic;

using System.Diagnostics.CodeAnalysis;

using System.Linq;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading.Tasks;


namespace Assignment2

{

    internal class Program

    {

        static void Main(string[] args)

        {



        }


        sealed class Bank

        {

            public int AccountNumber;

            public string name;

            public double balance;


            public Bank(int AccountNumber, string name, double balance)

            {

                this.AccountNumber = AccountNumber;

                this.name = name;

                this.balance = balance;
```

```csharp
        }
        public void deposit(double balance)
        {
            balance += balance;
        }
        public void withdrawl(double amount)
        {
            if (amount > balance)
            {
                Console.WriteLine("Insufficinent Balance");
            }
            else
            {
                balance = balance - amount;
            }
        }
        public void displayData()
        {
            Console.WriteLine("AccountNumber:" + this.AccountNumber);
            Console.WriteLine("Name :" + this.name);
            Console.WriteLine("Balance :" + this.balance);
            Console.ReadKey();
        }
    }
    class SavingAccount : Bank
    {
        public int AccountNumber;
        public string name;

    }
}
```

}