**DAY – 6 Assignment**

## Access Modifiers

**Public** - members were available anywhere in the program for all classes

**Protected** - members were only available to the same class

**Private** - members were only available to the inherited classes.

**Internal**- members can only be used in its own assembly, where assembly is the basic compilation unit of a program

**Internally Protected** - members can only be used in a particular assembly.

## **Property**

A property is a special type of combination of two methods “get” and “set”.

We use property when we have to provide access to a private member of a class.

This helps in encapsulation.

The “get” method is used for returning a value

While the “set” method is used for assigning a value to the field.

### **Syntax-**

## Polymorphism

Polymorphism means many forms and this can be achieved by Inheritance in classes, function overriding and function overloading

**Inheritance**

This is a feature of polymorphism in which one class can access the members of some other classes provided that the second class inherits the first class

For this purpose we have mainly 4 keywords which serves different purposes.

* **Virtual Keyword**

For making a member (function/property) to be able to inherited and can be overridden by some other class

* **Override Keyword**

If a class inherits a member of some other class then it has the privilege to re-define that function or property using override keyword.

* **New Keyword**

For hiding parent class property/ method we use new keyword

But the child class of this derived class can inherits the first base class's members

* **Sealed Keyword**

When we want to break the hierarchy of inheritance to the base c lass and don't want other classes to inherit from a class.

## **Constructor**

* A constructor is a special method which is used to initialize objects.
* It gets automatically called when an object is created and
* Has the same name as that of class and no return type.

1. **Default Constructor**

This constructor does not take any parameters

It initialized objects with some default value(like all numeric fields will be assigned zero)

A default constructor will automatically be called if no constructor is defined

1. **Parameterized Constructor**

A parameterized constructor is one which takes at least one parameter

1. **Copy Constructor**

Take reference of its own type object

It is used to create an instance of to the new values with existing instance,

One object is created by copying variables from another another object

1. **Static Constructor**

Static constructor is used by all the instances of class

It is created using the creation of the first object.

1. **Private Constructor**

**class Person**

**{**

**private string name; // field**

**public string Name // property**

**{**

**get { return name; }**

**set { name = value; }**

**}**

**}**

**class Program**

**{**

**static void Main(string[] args)**

**{**

**Person myObj = new Person();**

**myObj.Name = "Liam";**

**Console.WriteLine(myObj.Name);**

**}**

**}**

**class Person**

**{**

**public string Name // property**

**{ get; set; }**

**}**

**class Program**

**{**

**static void Main(string[] args)**

**{**

**Person myObj = new Person();**

**myObj.Name = "Liam";**

**Console.WriteLine(myObj.Name);**

**}**

**}**