INHERITANCE AND POLYMORPHISM

WHAT IS INHERITANCE?

Inheritance allows us to create new classes based on existing ones.

A new class (subclass, derived class, child class) inherits the fields, constructors, and methods of the class it is based on (superclass, base class, parent class).

EXTENDING CLASSES

A subclass can extend the superclass by adding new fields, constructors, and methods. It can also override a method from the superclass with its own version of the method.

Classes in C# can derive from only a single direct base class (Single Inheritance)

USING INHERITANCE

Create generic superclasses that implement common elements of related subclasses.

Create classes that inherit from classes that are defined by the .NET API.

ACCESS MODIFIERS

ACCESS MODIFIERS SPECIFY THE ACCESSIBILITY OF THE MEMBERS DECLARED BY A CLASS.

- Public: You can access from everywhere.
- Protected: Access is limited to within the class and any class that inherits from the class
- Private: Access is only limited to within the class

ACCESS MODIFIERS

- Internal: Access is limited exclusively to classes defined within the current project assembly
- protected internal: Access is limited to the current assembly and types derived from the containing class.

CREATING A SUBCLASS

SYNTAX

```
public class SubclassName:SuperclassName
{

}
```

CREATING A SUBCLASS

We can override methods coming from the parent class if they are defined as virtual.

The override keyword gives the programmer a chance to change the behaviour of methods passed from the parent class.

We can use the "base" keyword to call a constructor or method of the parent class.

CREATING A SUBCLASS

EXAMPLE ON OVERRIDE

```
class Person
    {  protected string Name;
        public virtual void PrintName()
        {
        Console.WriteLine(Name);
        }
    }
```

CREATING A SUBCLASS

EXAMPLE ON OVERRIDE

```
class Employee : Person
        {   private string Title;
            public override void PrintName()
                {
                    Console.WriteLine(Title +" " + Name);
                }
        }
```

POLYMORPHISM

WHAT IS POLYMORPHISM?

Polymorphism is a feature of inheritance that allows us to treat object of different subclasses that are derived from the same superclass as if they had the type of the superclass.

POLYMORPHISM

EXAMPLE ON POLYMORPHISM

Using the Person and the Employee class

```
Person p = new Employee();
```

SEALED CLASSES

We can prevent a class from being inherited by using the sealed keyword.

```
sealed class MyClass
{
// this class cannot be extended!
}
```

RECAP

What you should know at this point:

- What is Inheritance.
- Importance of Inheritance.
- How Inheritance works.
- What is polymorphism.
- Why we need polymorphism.
- Using the sealed keyword.