C#   
INHERITANCE

## Objective

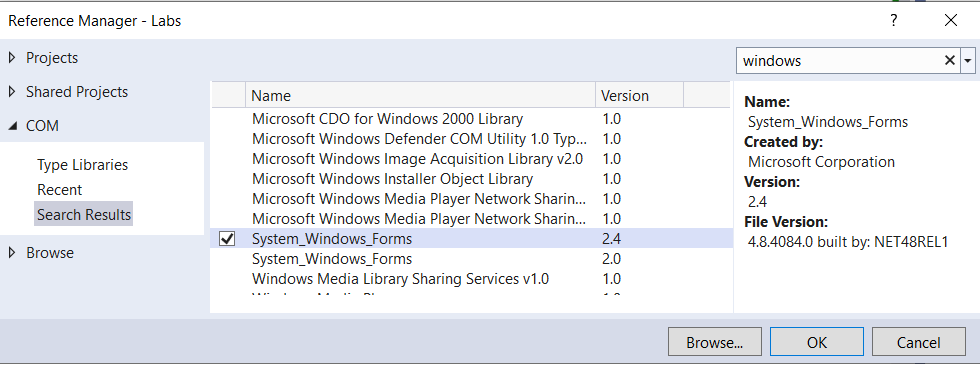
The primary objective for this lab is to enable you to derive new types and to add specialist functionality.

## Overview

The lab introduces some of the basic concepts of the inheritance story. As mentioned in the associated session, To implement inheritance, you must first have a class that provides the fundamental definition or behavior you need. In this lab we will play about with circular shapes.

### Step by step

1. Open the **Labs** Console project and make it the *startup project*.
2. The Console lab needs a couple of structures which are included in a Windows Form type of application but not in a Console app. Therefore, we will need to get the libraries for these structures.
   1. Right-mouse click on the Console project’s Dependencies in the Solution Explorer window
   2. Select ***Add Project Reference…***
   3. Click ***COM*** on the left-hand side (**C**omponent **O**bject **M**odel)
   4. Type ***Windows*** in the search box on the right
   5. Select (check the check box) **System\_Windows\_Forms**
   6. Click **OK**



1. Add the following classes using the instruction below.

Diagram

Description automatically generated

1. Create a constructor for Shape to set its Colour and Position
   1. ***Position*** is of type **Point** which is a class with built-in x and y.
   2. ***Colour*** is of type **Color**Both Color and Point can be resolved using System.Drawing;
2. As you can see, **Circle** extends **Shape** and **Sphere** extends **Circle.**

Tip: to calculate the Area/Circumference/Volume you will need **Math.PI** to get the value of **PI**.

Volume of an sphere is calculated as **4/3.0 \* PI \* R^3** (R to power of 3)  
You can use the **Math.Pow**(R, 3) fuction or R \* R \* R

Please note, Area, Circumference and Volume are defined as Property procedures. Just make sure the **Set** part does not exist by deleting it.

1. Create property procedures for each of the fields (colour, radius…) as indicated in the UML class diagram above.
2. The **GetDetails()** method returns a *String* containing all the attributes of the shape. It will be up to the caller how to display this information.
3. Create a new static method called Lab7() in the Program class.
4. Call Lab7() from within the Main() method
5. Create a few types of shapes in Lab7().
6. Print the characteristics of the Rectangle, Circle and Sphere objects which you've created.

## Create a List of Shapes

1. Create a **List<Shape>** called **shapes** in the Lab7() method.
2. Add the shapes which you created earlier into the *shapes* List.
3. Create a **foreach** loop to scroll through each shape and print its **colour** and **position**’s X and Y.  
     
   How does this work?!   
   How can we store a shape like Rectangle in a List of Shapes?

**All will be revealed in the next chapter!**

**\*\* End \*\***