Start a new Visual Studio console project, call it ShapesHW.

[1] Add a new ***public*** class called Shape

You can add this class in the same file as program.cs, or on its own file in the same project, or you can “do it right” and add a new project, a ClassLibray project, and add this Shape class there. We have done it all 3 ways so you can look at prior projects to make sure you get this correct.

Edit the Shape class:

Add a string property called Color

Add a virtual Print method. This method should contain just a Console.WriteLine statement that writes out “Hi from the parent Shape”

[2] Now add another new ***public*** class, called Square, it should inherit from the Shape class.

Add an int property called Size

Give it a Print method, which overrides the parents. Its Print method should print out a message that says:

“This is a ***color*** square with sides of size ***x***”

Where the value of ***color*** comes from its Color prop and the value of ***x*** comes from its Size prop.

[3] Now add another new ***public*** class, called Circle, it should inherit from the Shape class.

Add an int property called Size

Give it a Print method, which overrides the parents. Its Print method should print out a message that says:

“This is a ***color*** circle with radius of size ***x***”

Where the value of ***color*** comes from its Color prop and the value of ***x*** comes from its Size prop.

[4] Now add another new ***public*** class, called Rhombus, it should inherit from the Shape class.

Add an int property called Size

Give it a Print method, which overrides the parents. Its Print method should print out a message that says:

“This is a ***color*** Rhombus with sides of size ***x***”

Where the value of ***color*** comes from its Color prop and the value of ***x*** comes from its Size prop.

[5] Now in the Main method of your program.cs, instantiate 6 objects, and set their color and size prop’s. Instantiate 2 objects of type square, 2 objects of type circle, and 2 objects of type diamond.

[6] If your Program.cs does not have this, add at the top:

using System.Collections.Generic;

Create a new List of type Shape

List<Shape> myShapeList = new List<Shape>();

add your 6 objects to the myShapeList by doing 6 lines like (assuming you named your first square object **square1**)

myShapeList.Add(square1);

[7] Using a foreach loop, call the Print method of each of the 6 objects.

foreach (Shape item in myShapeList)

{

item.Print();

}

My code outputs this:

