# Form 1:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace SportsTeams

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

SoftBallTeam mySoftBallTeam = new SoftBallTeam(); //instantiate

//my method

//make a label or textbox equal your string

}

private void button2\_Click(object sender, EventArgs e)

{

BasketBallTeam myBasketBallTeam = new BasketBallTeam();

//my method

//make a label or textbox equal your string

}

private void button3\_Click(object sender, EventArgs e)

{

CrossCountryTeam myCrossCountryTeam = new CrossCountryTeam();

//my method

//make a label or textbox equal your string

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

}

}

# A Class That You Add Yourself:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SportsTeams

{

public abstract class BaseTeam

{

// Properties

// Default constructor.

public virtual string[] GetInfo() //is the method that override changes

{

//some code that compiles your properties into one array or string

//return statement

}

}

public class SoftBallTeam : BaseTeam

{

// Properties that are unique to the derived class

// Constructor

public override string[] GetInfo()

{

//some code that adds your new properties to the string/array

return strInfo;

}

}

public class BasketBallTeam : BaseTeam

{

// Properties that are unique to the derived class

// Constructor

public override string[] GetInfo()

{

//some code that adds your new properties to the string/array

return strInfo;

}

}

public class CrossCountryTeam : BaseTeam

{

// Properties that are unique to the derived class

// Constructor

public override string[] GetInfo()

{

//some code that adds your new properties to the string/array

return strInfo;

}

}

}

# Sample Output:

