# Overview

To continue working with Object-Oriented Synergy.

# Resources

* [Synergy DBL Language Reference](https://www.synergex.com/docs/index.htm)
* [System-Supplied Classes](https://www.synergex.com/docs/versions/v111/index.htm#lrm/10_System_Classes.htm)
* [Understanding Objects and Classes](https://www.synergex.com/docs/versions/v111/index.htm#lrm/8_Classes.htm)
* [Synergy Best Practices - Coding Standards](http://jobfunc2.cu.net/Job%20Functions/Programmer/Programmer%20Handbook/Tims%20Best%20Practices%20-%20Standards/Synergy%20Best%20Practices%20-%20Coding%20Standards.docx)
* [Traditional Synergy in Visual Studio - CU Wiki](http://echo.cu.net/cuwiki/Traditional_Synergy_in_Visual_Studio)
* [Traditional Synergy in Visual Studio Common Terminology - CU Wiki](http://echo.cu.net/cuwiki/Traditional_Synergy_in_Visual_Studio_Common_Terminology)
* [Installing Traditional Synergy in Visual Studio Templates - CU Wiki](http://echo.cu.net/cuwiki/Installing_Traditional_Synergy_in_Visual_Studio_Templates)
* [Creating a New ELB Project (TSVS) - CU Wiki](http://echo.cu.net/cuwiki/Creating_a_New_ELB_Project_(TSVS))
* [Debugging (TSVS) - CU Wiki](http://echo.cu.net/cuwiki/Debugging_(TSVS))

# Exercise

For this exercise you should complete the following steps in the order shown:

1. Create a Rooster class similar to a Hen.
2. Create a Chicken class that abstracts the common behaviors of Hens and Roosters.
3. Change the Hen and Rooster classes so they extend Chicken.
4. Modify unit tests to include tests for Rooster and verify existing Hen tests still pass.
5. Modify the Chicken class as follows:
   * Make the class abstract.
   * Make the RetailPrice a virtual method.
6. Modify the Rooster class as follows:
   * Add property Crows (Boolean) get/set.
   * Override RetailPrice so that it adds a premium of $0.50 if the rooster crows.
   * Add a unit test to verify this behavior.
7. Modify the Chicken, Hen, and Rooster classes as follows:
   * Add an abstract method to Chicken CalculateFoodConsumption:
     + It has no parameters.
     + Returns a decimal indicating the amount of food in pounds.
   * Implement CalculateFoodConsumption for Hen:
     + Weight \* 0.25
   * Implement CalculateFoodConsumption for Rooster:
     + Weight \* 0.10 (because Roosters eat less since they don't lay eggs).
   * Add unit tests to cover this method.

# Discussion

This is obviously a complex exercise, and the key to success is to take it slowly and step by step.