# ISIT 324 Homework 3

**30 Points Possible**

ACME Anvils Corporation uses some code in its pricing calculations that has some issues. The class in question is in a VS solution packaged as a .zip file linked to from this assignment. The methods inside the class include comments that describe how the method is intended to behave.



Questions:

1. (**10 points possible)** Describe the defects and what you must do to fix them (either in text or as code snippets) so that the program will work as described in the comments. You’re not looking to resolve ambiguities in this assignment; you’re looking for actual program defects. **There are at least 3 defects.** I don’t know of any that do not result in failures, but you might find something I missed…

**Method:** CalcPricePerAnvil

**Problem Line:** else if (quantity > 10 && quantity < 20)

**Defect:** If the quantity is set to 10, the method will erroneously return a regular price instead a 10% discount. This occurs because the else if statement only checks if the quantity is less than 10 rather that less than or equal to 10.

**Fix:** Change the else if statement to check for equal or less than 10.   
else if (quantity >= 10 && quantity < 20)

**Method:** CalcShippingCostPerAnvil

**Problem Line:** int[] zoneCost = new int[] { 10, 20, 30 };

**Defect:** The business logic does not take the 0 element of the array into account. This makes the zone input perpetually 1 off the desired result.

**Fix:** Either rewrite the business logic or add logic that subtracts 1 to the zone perimeter.

**Method:** CalcPricePerAnvil

**Problem Line:** n/a

**Defect:** The method does not correctly handle a negative regPrice parameter.

**Fix:** Introduce logic that checks for a negative parameter and return 0 if true.

1. (**20 points possible)** Correct the code and develop test cases that demonstrate that the defects have been corrected. To do so, create a VS Solution that includes:
   1. One project that holds the class with the corrected code, and
   2. One project with test cases to demonstrate that the code is fixed. This means that there must be at least one test case for every error state and failure that you find. All tests must pass.
   3. For the CalcPricePerAnvil method, use parameterized test methods. Include at least 5 DataRows.

Submit a text entry or a .doc file for question 1.

Zip up your Solution folder and submit the zip file for question 2.

**For full credit:**

* When I unzip the solution you submit, the tests must run. Make sure you zip up all the right stuff
* Use one test class for each of the two methods under test.
* Use the recommended name formulation for each of the test methods.
* Arrange your test methods using the “arrange, act, assert” pattern (including comments).
* Use the SUT alias to identify the software under test.
* Use at least one instance of “Assert.ThrowsException” in your test code.