Testing Practice

In each of these, list several tests, including parameters and expected results.

Do not attempt to code these functions! We will do that later. Also, do not write XUnit tests. Simply list in a notepad document the parameters and expected results.

WordJoin function

```
public static string WordJoin(string)
```

This function takes a single string parameter that contains words separated by a single space, such as "Hello there everybody".

It then combines the words together without spaces and capitalizes the first letter of each word. Thus "Hello there everybody" will become "HelloThereEverybody". The function will accept as a “word” any sequence of characters other than space. For words starting with non-alphabetic characters, no change for “uppercase” will be done. Thus the string “Abc %def !%” will return “Abc%def1%”. If either a null or empty string is passed in, the function will return null.

Coupon function

```
public static decimal CalculateCoupon(decimal purchase)
```

This function takes as a parameter the amount purchased, and calculates a coupon the customer should receive. When a customer purchases at least $10 up to and including $19.99, the customer will get a $2 coupon back. When a customer purchases $20 or more, the customer will get a $3 coupon back. If the customer purchases under $10, then no coupon will be given.

Thus, if the parameter is between 10.00 and 19.99 inclusive, the function will return 2.00.

If the parameter is 20.00 or higher, the function will return 3.00.

If the parameter is under 10.00, then the function will return 0.00.

Note: This function does not deal with tax calculations.

IntegersInString function

```
public static int[] IntegersInString(string nums)
```

This function takes as a parameter a single string consisting of integers separated by commas and returns an array of integers.

For example, the input string "10,3,6,0,-5,100" would return an array of integers consisting of 10, 3, 6, 0, -5, and 100.

Any string containing characters other than integers and commas will throw an exception of type System.FormatException. Thus a string such as “Hello,1,5” would throw FormatException.

IsFactor function

```
public static bool IsFactor(int X, int Y)
```

This function takes two parameters and determines if the second is a factor of the first. If the second is a factor of the first, it returns true. If not, it returns false.

Tip: For this final exercise, think carefully about the situation if the second parameter is zero. Technically, the specification does indeed say what to return when the second parameter is zero.