# ISIT 324 Homework 2

**30 Points Possible**

1. Given your solution to the Module 1 Homework, question 3:
   1. You may discover a variety of ambiguities in the assignment. In other words, ample opportunities for defects exist. Describe as many possible defects as you can. *Submit either a text file or a Word doc.* **(8 points)**

**\*Returns incorrect data**

**\*Accepts the wrong data type**

**\*Returns the wrong data type**

* 1. Create a set of test cases that you think would have a reasonable chance of revealing the faults you identified above and document the rationale in your test set as inline comments. (Note: you don’t have to use a test framework for this. Just put your test code in Main() and call the Union method for each test. Print “test passed” or “test failed” for each test. *Submit a .cs file including the Main() method with tests and the Union method.* **(12 points)**
  2. (recommended) Run your tests with your implementation to ensure you get the expected results.

1. Given the following code (note that the program is written in Java and doesn’t follow C# conventions), answer the questions below. You might note that it’s very similar to question 2 in the Module 1 assignment. **(10 points)**

|  |
| --- |
| /\*\* |
| \* Count positive elements |
| \* |
| \* @param x array to search |
| \* @return count of positive elements in x |
| \* @throws NullPointerException if x is null |
| \*/ |
| public int countPositive (int[] x) |
| { |
| int count = 0; |
| for (int i=0; i < x.length; i++) |
| { |
| if (x[i] >= 0) |
| { |
| count++; |
| } |
| } |
| return count; |
| } |
| // test: x = [-4, 2, 0, 2]; Expected = 2 |

* 1. Explain what is wrong with this code. Describe the fault precisely by proposing a modification to the code.

**The if statement in the for loop as written is including 0’s as a positive number. Because in the if statement, it is searching for numbers that are greater than or equal to 0 when it should be searching for numbers only greater than 0.**

* 1. Give a test case that does not execute the fault and explain why it doesn’t.

**// test: x = [ ]; Expected = 0**

**// test: x = null; Expected = Null Reference Error**

**Either an empty or null array passed in will not execute the defect. The for loop will not go into its body with an empty array. Whereas the null value will error out before executing any of the method.**

* 1. Give a test case that executes the fault but does **not** result in a failure.

**// test: x = [-4, 2, 1, 2]; Expected = 3**

* 1. Implement your fix to the code in C# and call it from Main() with your test cases to demonstrate that the fault is corrected. Then submit your .cs file.