# ISIT 324 Homework 1

Submission should include one Word/PDF doc and two .cs files.

1. Explain the terms **error, defect** and **failure** and provide an example of each**.** Don’t just repeat the definitions/examples from the book or lecture.

An **error** is created when a human makes a mistake that causes an issue. Such as, entering in inaccurate value or including a typo.

Errors led **defects.** Defects are problems or deficiencies in the code or work that compromises the final product.

Defects led to **failures.** Failures are the real time results of defects. They occur when actively using a product with defects in it.

1. Given the following code (note that the program is written in Java and doesn’t follow C# conventions), answer the questions below.

|  |
| --- |
| /\*\* |
| \* Find last index of element |
| \* |
| \* @param x array to search |
| \* @param y value to look for |
| \* @return last index of y in x; -1 if absent |
| \* @throw NullPointerException if x is null |
| \*/ |
| public int findLast (int[] x, int y) |
| { |
| for (int i=x.length-1; i > 0; i--) |
| { |
| if (x[i] == y) |
| { |
| return i; |
| } |
| } |
| return -1; |
| } |
| // test: x = [2, 3, 5]; y = 2; Expected = 0 |

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

The defect occurs in the for loop. Because the loop never reaches the 0 element of the array because it ends when i is greater than 0. In other words, the loop ends when i = 0 and before it can process the 0th element.

* 1. Give a test case that does not execute the defect (tricky!)

// test: x = [2, 3, 5]; y = 3; Expected = 1

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

// test: x = [2, 3, 5]; y = 6; Expected = -1

* 1. Implement your fix to the code in C# and call it with test cases from **Main()** to demonstrate that the faults are corrected. Submit your .cs file.

1. Write a C# method with this signature and submit the .cs file:

         public static ArrayList Union(ArrayList a, ArrayList b)

The method should return a list of objects that are in either of the two argument vectors.

You’ll use this program to answer questions in a subsequent exercise.