Nathan VelaBorja
December 7, 2016
CptS 422 – HW 14

1.

Index j is not decremented in the while loop, so for each number in the list, it could only be swapped left a maximum of one position. To fix this, simply add "j--" to the end of the while loop.

2.

This problems throws a null reference exception, because reflection is unable to see the method "Function". Since Function() doesn't specify access level, it defaults to internal. To fix this, just set the access modifier of Function() to public.

3.

```
float f1 = 4.25f;        // Lossless. 0.25 can be represented perfectly by a power of 2
double d1 = f1;          // Lossless. The double simply adds exponent and fraction bits
float f2 = 1.4f;         // Lossy. A float cannot perfectly represent 0.4 using powers of 2
double d2 = f2;          // Lossless. Same as 2. The float f2 is lossy, but casting it to a double
                                 won't lose any data.
```

4.

When we create a reference type for that int, we initialize it to the value of num, creating sort of a wrapper class. However, the original num is still a value type and when it is updated, the reference object is not affected.

5.

This code is a good start, but the goal is to determine of the HTTP method is valid or not. Here, only the very beginning of the method is checked, but we need more information to verify it is in fact valid. This code would accept "GET out of my house you monkey", when that is obviously not valid.

6.

```
int w = a + b;           // Lossy. Ex: If a is int.Max and b > 0, w will not be correct
int x = a – b;           // Lossy. Ex: if a is int.Min and b > 0, x will not be correct
int y = a * b;           // Lossy. Ex: if a is int.Max and b > 1, x will not be correct
int z = a / b;           // Lossy. Ex: if a = 10 and b = 3, z will lose precision and display 3.
```

7.

```
1. bool method1 = File.Exists(filePath);
2. bool method2 = (File.Open(filePath, FileMode.Open, FileAccess.Read) != null);
3. bool method3 = (new FileStream(filePath, FileMode.Open, FileAccess.Read)) != null;
```

Method 1 is the most efficient, as it simply uses the File class to search through directories and see if the entry exists. Methods 2 and 3 work, but they are less efficient since they both open streams to those files for no reason just to check if the file exists. Also, they could potentially fail even if the file does exist if the file is already opened with Write permissions.

8.

The problem with the dynamic object creation for something like a Linked List Node is that the properties of the dynamic object cannot be changed once they are constructed. This limitation is detrimental for most advanced implementations, including Linked Lists.

9.

There are likely many ways to accomplish this in C#, but one I found is a class called InterLocked. This controls thread access to a shared variable and makes sure everything goes smoothly using some sophisticated magic. It could be used right in the increment line.

10.

string stringValue = “\0”;

I chose this because I google “How to break ToString()” and it came up.
After looking into the way xml handles input, it seems that xml has a range of invalid characters that can't be put in the content of a tag described by hex values, so I'm guessing the \0 encodes to some value in that range of unacceptable values.