Junaid Khan
(Jay)
11346814

1. put j -= 1 at the end of the nested while loop. This is because j isn't decrementing the entire time.

I understand why this question is so important... As small and simple of an oh “Gotcha” this answer was, it really tests a crucial skill that most of us graduating have to remember. Don't overlook anything, test EVERYTHING.

2. GetMethod can only get public methods, the simple fix is to make void Function a public method.

3.
```
float f1 = 4.25f; is exactly 4.25
double d1 = f1; lossless due to IEEE754
float f2 = 1.4f; is not exactly 1.4
double d2 = f2; lossless due to IEEE754
```

4. This is called boxing (Just had a flashback to Evan's awesome 322 class I took 1.5 years ago :-)) All we're doing here “boxing” the value of the value type, num -into an Object or wrapping an Object around it and so if num is incremented it wont affect o. So num is 42, and o is 41.

5. firstLine can start with valid but after that it may not follow “SP RequestURI SP HTTPVERSION CRLF”.
I.e it can be something like GETBLAHBLAH\r\n which is invalid, but we would never know due to only looking for if the string starts with one of those validMethods. This is not sufficient enough to achieve our goal of validation.

6. If we are not counting overflow as lossy (as I understand it, I'm not 100% sure though)
i) Lossless, it will add and retain the value and will overflow as need. (I tested MaxValue)
ii) Lossless, it will subtract and retain the value and will overflow as need (I tested MinValue)
iii) Lossy
iv) Lossy, since it just trims off it's numbers after decimal point.

7. method1 as per msdn: If the caller does not have sufficient permissions to read the specified file, no exception is thrown and the method returns **false** regardless of the existence of *path*. This is inferior in that it's not even a method that it cannot tell us if the path exists for sure. What if it does exist? We will never truly know.

Method2: This is inferior to method3 in that there is extra overhead (although very small), it is just a static method that returns method3 .

Method3: superior, will throw security exception if insufficient privileges. It is what method2 is calling so less overhead.

8. Because of using anonymous type nodes, you essentially have a readonly singly linked list in terms of not only can the nodes not update their values but the linkde list cannot insert any nodes starting from the root (I.e in the middle and at the end). I read the msdn documentation for this.

Junaid Khan
(Jay)
11346814

9. To make it thread-safe, change the line:

```
for(int i = 0; i < 100000; i++) { s_sharedValue++; }
```

to

```
for (int i = 0; i < 100000; i++) { Interlocked.Increment(ref s_share
dValue); }
```

What Interlocked.Increment does is it increments s_sharedValue then stores that result as an atomic operation. I only knew about this because I had to use something similar during my internship at Expedia this past summer (but in Java). Because of that, my first instinct was to actually use this as opposed to a lock.

10. I first tried putting in unicode and that didn't throw an exception, so I went to the ASCII chart and wondered how odd ASCII chars (by odd I mean not showing up on a standard keyboard) like NUL would be represented in xml, and I know you can escape them in a string, so it threw an exception as I expected and they are in fact invalid characters in XML.