

Add Length property to FailSoftArray

```
/*
```

```
C#: The Complete Reference
```

```
by Herbert Schildt
```

```
Publisher: Osborne/McGraw-Hill (March 8, 2002)
```

```
ISBN: 0072134852
```

```
*/
```

```
// Add Length property to FailSoftArray.
```

```
using System;
```

```
class FailSoftArray {
```

```
int[] a; // reference to underlying array
```

```
int len; // length of array – underlies Length property
```

```
public bool errflag; // indicates outcome of last operation
```

```
// Construct array given its size.
```

```
public FailSoftArray(int size) {
```

```
a = new int[size];
```

```
len = size;
```

```
}
```

```
// Read-only Length property.
```

```
public int Length {
```

```
get {
```

```
return len;
```

```
}
```

```
}
```

```
// This is the indexer for FailSoftArray.
```

```
public int this[int index] {
```

```
// This is the get accessor.
```

```
get {  
    if(ok(index)) {  
        errflag = false;  
        return a[index];  
    } else {  
        errflag = true;  
        return 0;  
    }  
}
```

// This is the set accessor

```
set {  
    if(ok(index)) {  
        a[index] = value;  
        errflag = false;  
    }  
    else errflag = true;  
}
```

// Return true if index is within bounds.

```
private bool ok(int index) {  
    if(index >= 0 & index < Length) return true; return false; } }  
// Demonstrate the improved fail-soft array. public class  
ImprovedFSDemo { public static void Main() { FailSoftArray fs  
= new FailSoftArray(5); int x; // can read Length for(int i=0;  
i < fs.Length; i++) fs[i] = i*10; for(int i=0; i < fs.Length;  
i++) { x = fs[i]; if(x != -1) Console.Write(x + " "); }  
Console.WriteLine(); // fs.Length = 10; // Error, illegal! } }  
[/csharp]
```