

Use an indexer to create a fail-soft array

```
/*
C#: The Complete Reference
by Herbert Schildt

Publisher: Osborne/McGraw-Hill (March 8, 2002)
ISBN: 0072134852
*/

// Use an indexer to create a fail-soft array.

using System;

class FailSoftArray {
int[] a; // reference to underlying array

public int Length; // Length is public

public bool errflag; // indicates outcome of last operation

// Construct array given its size.
public FailSoftArray(int size) {
a = new int[size];
Length = size;
}

// 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 fail-soft array. public class FSDemo {
public static void Main() { FailSoftArray fs = new
FailSoftArray(5); int x; // show quiet failures
Console.WriteLine("Fail quietly."); for(int i=0; i <
(fs.Length * 2); i++) fs[i] = i*10; for(int i=0; i <
(fs.Length * 2); i++) { x = fs[i]; if(x != -1) Console.Write(x
+ " "); } Console.WriteLine(); // now, generate failures
Console.WriteLine(" Fail with error reports."); for(int i=0; i
< (fs.Length * 2); i++) { fs[i] = i*10; if(fs.errflag)
Console.WriteLine("fs[" + i + "] out-of-bounds"); } for(int
i=0; i < (fs.Length * 2); i++) { x = fs[i]; if(!fs.errflag)
Console.Write(x + " "); else Console.WriteLine("fs[" + i + "]
out-of-bounds"); } } } [/csharp]

```