Use a custom Exception for RangeArray errors

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

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

// Use a custom Exception for RangeArray errors.

using System;

// Create an RangeArray exception.
class RangeArrayException : ApplicationException {
// Implement the standard constructors
public RangeArrayException() : base() { }
public RangeArrayException(string str) : base(str) { }

// Override ToString for RangeArrayException.
public override string ToString() {
return Message;
}
}

// An improved version of RangeArray.
class RangeArray {
// private data
int[] a; // reference to underlying array
int lowerBound; // lowest index
int upperBound; // greatest index

int len; // underlying var for Length property

// Construct array given its size.
```

```
public RangeArray(int low, int high) {
high++;
if(high <= low) { throw new RangeArrayException("Low index not less than high."); } a = new int[high - low]; len = high - low; lowerBound = low; upperBound = --high; } // Read-only Length property. public int Length { get { return len; } } // This is the indexer for RangeArray. public int this[int index] { // This is the get accessor. get { if(ok(index)) { return a[index - lowerBound]; } else { throw new RangeArrayException("Range Error."); } } // This is the set accessor. set { if(ok(index)) { a[index - lowerBound] = value; } else throw new RangeArrayException("Range Error."); } } // Return true if index is within bounds. private bool ok(int index) { if(index >= lowerBound & index <= upperBound) return true; return false; } } // Demonstrate the index-range array. public class RangeArrayDemo1 { public static void Main() { try { RangeArray ra = new RangeArray(-5, 5); RangeArray ra2 = new RangeArray(1, 10); // Demonstrate ra Console.WriteLine("Length of ra: " + ra.Length); for(int i = -5; i <= 5; i++) ra[i] = i; Console.Write("Contents of ra: "); for(int i = -5; i <= 5; i++) Console.Write(ra[i] + " "); Console.WriteLine(" "); // Demonstrate ra2 Console.WriteLine("Length of ra2: " + ra2.Length); for(int i = 1; i <= 10; i++) ra2[i] = i; Console.Write("Contents of ra2: "); for(int i = 1; i <= 10; i++) Console.Write(ra2[i] + " "); Console.WriteLine(" "); } catch (RangeArrayException exc) { Console.WriteLine(exc); } // Now, demonstrate some errors. Console.WriteLine("Now generate some range errors."); // Use an invalid constructor. try { RangeArray ra3 = new RangeArray(100, -10); // Error } catch (RangeArrayException exc) { Console.WriteLine(exc); } // Use an invalid index. try { RangeArray ra3 = new RangeArray(-2, 2); for(int i = -2; i <= 2; i++) ra3[i] = i; Console.Write("Contents of ra3: "); for(int i = -2; i <= 10; i++) // generate range error Console.Write(ra3[i] + " "); } catch (RangeArrayException exc) { Console.WriteLine(exc); } } } [/csharp]
```