An Auto-Adjusting Data Structure, Reading

The *ultimate* would be to have a data structure that could *detect* when its capacity needs to be increased. For example, if **Array<int> a(10);** is declared and we do **a[i]=100;** where "i" is 20, rather than let the operator[ ] setter fail and return the dummy, why not allow it to accommodate the value at index 20 by automatically adjusting its capacity?

So rather than this in the setter:

```
if (index < 0) return dummy;
if (index >= cap) return dummy;
```

...we could do this:

```
if (index < 0) return dummy;
if (index >= cap) capacity(2 * index); // more than enough to include this index
```

This removes the guesswork from choosing an initial capacity and managing it with the Array::capacity setter. In fact, the programmer using the Array can just accept its default (**Array<int> a;**), and not worry about upper limits!