Range Checking

Programmers coming from other languages often gravitate to the subscript operator since it is similar in syntax to the array operations which the vector emulates. However, if you use the subscript operator, your program will **do no range checking**!

Before you go any further, **read over that last line again**. Most of you probably **cannot imagine** a language that does not do some sort of range checking. Let me illustrate what happens in C++:

```
vector<int> v(4);   // 4 elements
cout << v[4];       // access out of bounds
v[4] = 27;          // writing out of bounds
cout << v[4];
```

You **will not get a compiler error or a runtime error**, as you would in Java or Python, even though you are accessing (and even writing to) an element that is outside of the vector bounds.

This is an error, though. Often, `cout` will print the value stored in the location where the fifth element of `v` **would be stored**, if it existed. If that is the case, on your platform, then the assignment will happily store the value `27` in the same location, **regardless of what is currently stored there**. If you think, "Well, that's not so bad!", then what about this?

```
v[1075935] = 27;
```

Again, you'll get **no nice stack trace** like you would get with Java or Python, telling you that your index is out of bounds. If you **are very lucky**, your operating system will shut down your application rudely with a **segmentation fault**. If you aren't, you will **silently corrupt a portion of your own application**, producing a bug that shows up days, weeks or months later. **Not good**.

Using the *at* Member Function

Fortunately, you can fix this deficiency by using `at()`. When you use `at()`, the compiler generates code to check out-of-bound subscripts; you don't have to rely on accidentally stepping on the toes of your operating system to find your errors.

Other than the slight performance hit, I can't think of any reason not to **always** use `at()` instead of the subscript operator. Combined with C++ **exception handling**, your code will be safer and have fewer bugs.

You can also modify the vector class so that subscripts do throw exceptions. That's what Bjarne Stroustrup does in Section 4.4.1.2 of the Tour of C++ (page 104).