C-Style Strings

C does not have a true "string" data type. Instead, it uses arrays of char. Since the strings are arrays, only legal array operations are allowed.

"Illegal" Operation: How to do it:

```c
s = "string"; strcpy(a, "string");
Can't assign a string (except in declarations)
```

```c
b = a; strcpy(b, a);
Can't copy one string to another
```

```c
if(a == "string")....
if(strncmp(a, "string") == 0)...
if(a < "string")...
if(strcmp(a, "string") < 0)...
if(a >= "string")...
if(strcmp(a, "string") >= 0)...
Can't compare strings with ==, <, >, etc.
(Actually, this isn't illegal, but it probably doesn't do what you want)
```

---

C++ string Class

A string class has recently been added to C++. It provides a convenient way to handle strings.

<table>
<thead>
<tr>
<th>&quot;C-style&quot; strings:</th>
<th>C++ string class</th>
</tr>
</thead>
</table>
| ```c
#include <cstring> // <string.h> in C
char a[50];
Must allocate enough space!
strcpy(a, "string");
strcpy(b, a);
``` | ```c
#include <string>
string a;
Dynamic allocation
``` |
| String assignment/copying | String comparison |
| ```c
if(strcmp(a, "string") == 0)...
if(strcmp(a, "string") < 0)...
``` | ```c
if(a == "string")....
if(a < "string")...
``` |
| String comparison | String concatenation |
| ```c
a = strcat(a, b);
``` | ```c
n = strlen(a);
``` |
| Length of string | ```c
n = a.length();
``` |
| ```c
ch = a[i];
The single character at position i
``` | ```c
ch = a.at(i)
``` |