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:*
`a = "string";`	`strcpy(a, "string");`
Can't assign a string (except in declarations)	
`b = a;`	`strcpy(b, a);`
Can't copy one string to another	
`if(a == "string")....`	`if(strcmp(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!)	

STRNG010

University of Idaho

C++ string Class

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

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

STRNG020

University of Idaho