COSC 301: Friday, 31 August 2012

Control structures

All the standard control structures from Java are valid in C: `while`, `for`, `if`, `do { ... } while ()`.

Again, there's no true boolean type in C. Beware. What's the output of this program and why?

```
int x = 0;
if (x = 10) {
    printf("Java is for coffee");
} else {
    printf("not programs!");
}
```

Arrays and strings

Arrays in C are similar to arrays in Java; multidimensional arrays also work:

```
int a[10];  /* this could be java... */
            /* these are standard C comments  */
int i;
for (i = 0; i < 10; i++) {
    a[i] = 0;
}
```

Note well: accesses to arrays in C **are not bounds checked!**:

```
int a[8];
int end = 12;
int i;
int v = 1;
for (i = 0; i <= end; i++) {
    a[i] = v;
    v *= 2;
    printf("at %d we have %d\n", i, a[i]);
}
```

What's the behavior of the above program?

Strings: there is no "real" built-in string data type. Instead, C uses arrays of `char` (in other words, a contiguous block of `char``s in memory). The string **must** be terminated with a ``NUL` byte: *'0'*.

String literals in double quotes, single characters in single quotes

Since arrays are not bounds-checked, you need to be careful when manipulating strings; also need to be careful to terminate them.

```
char buffer[64]; // space for 63 characters plus NULL char
strcpy(buffer, "hello, world");  // man strcpy;
      // what does array look like now?
buffer[8] = 'u';  // now?
buffer[12] = '!'; // and now?
```

There are many functions in the "C standard library" for string manipulation: `strcpy`, `strcmp`, `strlen`, `index`, etc. See `man` pages.

Example: let's reinvent `strlen`...