Lecture 14

Ch 9.1
Topics

• C-string
  – Null character
• Declaring a C-string
• Initializing C-string variables
• Index variables
• Library to include
• Assigning C-string value
  – strcpy
  – strncpy safe way
• Testing C-strings for equality
• More useful functions
• Fun with cin and cout
• getline
C-string

• What is it?
  – An array of chars
  – Must end with ‘\0’ otherwise it’s just an array of chars
‘\0’

• What is it?
  – The null character

• What does that mean?
  – This is a special representation of a character that is used to tell functions that the string has ended
Declaring a C-string

• char s[10];
  – How many characters can this have?
• char s[15] = “Over 9000!”;

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>v</td>
<td>e</td>
<td>r</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>!</td>
<td>\0</td>
<td>?</td>
<td></td>
</tr>
</tbody>
</table>
Initializing C-string variables

- char cowSays[] = “Moo!”;
  - Same as
- char cowSays[5] = “Moo!”;

- Is this the same?
  - char cowSays[] = {’M’,’o’,’o’,’!’};
Index variables

char cowSays[] = "Moo!";
cowSays[0] = 'M';
cowSays[1] = 'e';
cowSays[2] = 'o';
cowSays[4] = 'w';

What do cows say now?
Index variables

//How to mute a cow
char cowSays[MAX] = “Moo!”;
int index = 0;
while((cowSays[index] != ‘\0’) && (index < MAX))
{
    cowSays[index] = ‘ ‘;
    index++;
}
Library to include

• Well you do not need any libraries to use C-strings
• However there are many functions that manipulate C-strings for you
• These functions are in <cstring>
Assigning C-string value

This is how you assign a C-string

```c
char iAm[10];
iAm = “Trolling you”;
```
Assigning C-string value

• Seriously now
  – char iAm[20] = “Not trolling you”;

• Can only be assigned when declared in this way.

• To assign it later you must use string copy
  – strcpy(iAm, “Not trolling you”);

• This version of string copy doesn't check the length of the string.
Assigning C-string value

• To be safe use `strncopy`
  – `char size10[10]`;
  – `strncopy(size10, "String longer than 9", 9);`
Testing C-strings for equality

- Can’t do
  - if( cString1 == cString2)
  - Sorry
- Use strcmp or strncmp
  if(strcmp(cString1, cString2))
    cout << “Not the same”;  
  else
    cout << “They are the same”;  
- It compares the numerical value at each array position of the two arrays.
  - If the first is smaller it returns -1, if bigger it returns 1.
# More useful functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>strcat(str1,str2)</code></td>
<td>Concatenates <code>str2</code> to the end of <code>str1</code></td>
<td>This version doesn't check a limit.</td>
</tr>
<tr>
<td><code>strncat(str1,str2, limit)</code></td>
<td>Concatenates <code>str2</code> to the end of <code>str1</code> but it will do at most (limit) character</td>
<td>This version is safer if the limit is chosen well.</td>
</tr>
<tr>
<td><code>strlen(str1)</code></td>
<td>Returns the length of <code>str1</code>. So number of chars up until but not including ‘\0’</td>
<td></td>
</tr>
</tbody>
</table>
Fun with cin and cout

• char a[80], b[80];
• cout << “Enter some input:\n”;
• cin >> a >> b;
• cout << a << b << “END OF OUTPUT\n”;

• What will the result be?
getline

- char a[80]
- cout << “Enter some input\n”;
- cin.getline(a, 80); //Works on input streams
- cout << a << “END OF OUTPUT\n”;