CS2141 – Software Development using C/C++

Strings
Character functions

• The old cctype header defines some functions for working with characters:
  • isalpha( c ) True if c is alphabetic
  • isupper( c ) True if c is uppercase
  • islower( c ) True if c is lowercase
  • isdigit( c ) True if c is a decimal digit
  • isalnum( c ) True if c is alphanumerical
  • isspace( c ) True if c is whitespace
  • ispunct( c ) True if c is a punctuation character

• There are a few others
String Literals

- In C++ string literals are arrays of characters
- Usually a pointer is used to refer to the string
- These strings are null-terminated, meaning the last character in the array is the null character

```cpp
char * text = "A string literal";
```
String Literals cont.

• Pointers can be used to pass and manipulate string values:

```c
int vowelCount( const char * p ) {
    int sum = 0;
    while( 1 )
        switch( *p++ ) {
            case '\0': return sum;
            case 'a': case 'e': case 'i':
            case 'o': case 'u':
                sum++;
                break;
        }
    return sum;
}
```
cstring

- String functions from C are often used in C++
- These functions manipulate arrays of characters
- Often they assume the string is null-terminated
- Some of the functions provided by `cstring` (or `string.h`) are:
  - `strcpy( dest, src )`
    Copies characters from src to dest
  - `strncpy( dest, src, n )`
    Copies n characters from src to dest
cstring cont.

- `strcat(dest, src)`
  Append characters from src to dest

- `strncat(dest, src, n)`
  Append only n characters

- `strcmp(s1, s2)`
  Compare two strings

- `strncmp(s1, s2, n)`
  Compare first n characters of two strings

- `strlen(s)`
  Length of the string
C++ Strings

• C++ has a newer `string` type:

```cpp
#include <string>
using std::string;
...

string a;
string b = "Initial string";
string c( "Another string" );
string d( b );

a = "A different string";
```
The \texttt{string} type has many functions:

\begin{itemize}
  \item Number of chars \hspace{1cm} \texttt{s.length()} \\
  \item Assign \hspace{1cm} \texttt{s1 = s2} \\
  \item Append \hspace{1cm} \texttt{s1 += s2} \\
  \item Concatenate \hspace{1cm} \texttt{s1 + s2} \\
  \item Character access \hspace{1cm} \texttt{s[index]} \\
  \item Comparison \hspace{1cm} \texttt{s1 == s2, s1 != s2} \\
  \hspace{1cm} \texttt{s1 < s2, s1 > s2} \\
  \item Substring \hspace{1cm} \texttt{s.substr(start, length)} \\
  \item Input \hspace{1cm} \texttt{cin >> s;}
\end{itemize}
String Streams

• String concatenation and assignment only works with other strings

```cpp
int a = 5;
string text = "a=" + a;  // This won't work!
```

• For other data types, a string stream can be used:

```cpp
#include <sstream>
using std::ostringstream;
...
int a = 5, b = 6;
ostringstream formatter;
formatter << "The sum of " << a << " and " << b << " is " << (a + b);
string s = formatter.str( );
```