Lecture 03 Notes

Derek Harter

CSci 515 Spring 2014 *<2015-01-28 Wed>*

1 First Session (11 - 11:40)

1.1 Streams

The IOStream library is a new (object oriented) library, added with the C++ language, to support Input and Output to source and destination devices.

The source of input can be a keyboard, a file, or some other device. Likewise the destination of output can be to a file, to a terminal screen, or to some other device (for example you can send output into another C variable, like a string in memory).

A stream is a way of visualizing how data is transferred from the source to destination. A stream is inherently serial, the order in which you put things into the stream, is the order they will be received when they come out of the stream.

1.2 iostream header

You've already seen many examples of specifying the iostream header using

```
#include <iostream>
```

Iostream operators and objects are defined in the std namespace, thus you explicitly have to specify `std::` before using them, or include the

```
using namespace std;
```

directive.

In addition to iostream, if you want to do I/O to files, you need to include the fstream header. If you want to manipulate and format the data in/out of the stream, you need to include the iomanip header.

1.3 Standard Stream Objects

- cin, cout input from the standard input device, and output to the standard error device respectively. These are the keyboard and terminal, by default, but can be connected to others (like a file) by the OS, and program doesn't know or care.
- cerr send output to the standard error device, can be useful for separating error messages from normal output (and redirecting standard error to a different location). By default, standard error also goes to the terminal.
- clog also connects to the standard error output in a buffered manner. You don't need to be concerned with clog in this class.

1.4 Stream Output and Input

- using the << >> stream notation

```
cout << x << y << z;
cin >> x << y << z;
```

- Using member functions. The streams cout, cin, are objects, they have member functions. For example, and put and get single characters:

```
cout.put('A').put('\n');
cin.get(c);
```

- Example of reading a character at a time of input and echoing until EOF

```
int c; // use int, because char cannot represent EOF
while ( (character = cin.get()) != EOF)
{
  cout.put(character);
}
```

- Example of reading input a line at a time

```
const int SIZE = 80;
char buffer[SIZE];

cin.getline(buffer, SIZE);
```

- peek, putback and ignore can be used for low level I/O. We can ignore a number of characters, and we can peek ahead (without reading) or putback a character into the stream.

2 Second Session (11:45 - 12:30)

3 Third Session (12:40 - 1:40)