When you turn in the lab

Code (phsyical copy)

Show that it compiles

Show that it runs (make a main program)

* Streams
  + #include <fstream>
  + Ifstream – strictly input
  + Ofstream – strictly output (new file)
  + Fstream – both (random access)

Ifstream infile; // declare file handle

Infile.open("input.txt");

Infile >> variable;

Ofstream outfile; // declare the file handle

Outfile.open("valuable file. Do not erase txt."); // This file is clobbered

Outfile << variable;

Outfile.close();

Infile.close();

>> - input (white space is the delimiter)

#include <fstream>

Int main()

Fstream iofile;

Iofile.open("file.txt", flags);

// Flags ios::in – input

// ios::out – output

// ios:: in | ios:: out – simutaneously used for input and output

// ios::app – output to end of file

// ios::trunc – output after a nice clobber

// ios::binary – outputs the code

NOTE:

Iofile.open(" aishdashd", ios::out) //clobbered

Iofile.open(" ashdad'k", ios::in|ios::out) //keeps from being truncated

String file = "myfile.txt";

Iofile.open( file, Ios::in); //This will create errors, the file name is a c-string;

//You have to put this

Iofile.open(file.c.str(), ios::in); // c.str takes a C++ string and returns a c string

File IO operators

<< - output

>> - input

Getline - input.

Getline(file handle, string variable)

IE: string line;

Getline(iofile, line); //Will read all characters up to end of line. Does not save the EoL character – Will not be using this for this class.

Get/put - io one character at a time.

Char c = iofile.get();

Char c; iofile.get(c);

Char c = 'A'; iofile.put(c);

An open file has 2 pointers

P - Put (output)

G - Get (input)

When you open a file, both pointers are pointing the same character.

Iofile.peek will allow you to "peek" at the current location of G, without moving the pointer

Flags

Functions to indicate io flags on a file handle

Iofile.clear(); //resets the flags

Iofile.good(); //"everything is wonderful"

Iofile.bad(); || iofile.fail(); //"Something went wrong", hit eof

Iofile.eof(); // End of File is hit

Iofile.seekg(recordnumber\*n); //Will jump to a specified location on a record and place g pointer to it

Iofile.seekp(recordnumber\*n); //Will allow you to overwrite the record at this location

Int ipos = iofile.tellg(); // gives the position of the input pointer g

Int opos = iofile.tellp(); // gives the position of the output pointer p

Writing a little program – its in linux cse461 under April 5th