C++ file streams

C++ method to implement file input and output
file – named area of secondary storage used to hold information
file stream variable – a variable that represents a file inside a program

To use file I/O

- include the fstream header file
- declare variables to represent files -
  ifstream – data type for input files
  ofstream – data type for output files
- prepare (open) each file for reading or writing (2 methods shown below)
  - input file must exist or error occurs when first attempt to read is made
    - if not sure that input file exists, test status of ifstream variable to determine if file was successfully opened
  - output file will be created (if it does not exist) or prior content overwritten
- access the content of files by specifying variable name in each input or output statement
  - name of ifstream or ofstream variable is substituted for cin/cout
  - input and output operators (>> <<), I/O functions, output manipulators will all work with filestream variables as they do with cin/cout
- parameter passing – filestream parameters MUST be passed by reference
- when operations on a file are complete, close the file

C++ file streams

```
#include <fstream>  //inclusion of header file for use of filestreams
#include <iostream>
#include <string>   //need string variable to store file name
using namespace std; //~lee/cs135sampledir/samplefile.cpp
int main()
{
  ifstream infile;  //variable representing input file
  string filename;  //name of input file
  ofstream outfile; //variable representing output file
  int n;
  cout << "Please enter name of input file\n";
  cin >> filename;
                                 //open file when name
  infile.open(filename.c_str()); //stored in string
  outfile.open("myresults");     //open a file with a fixed name

  if (!infile) //test if infile is open for reading
     cout << “file not opened for input\n”;
  else   // file is open for reading
   {
     infile >> n;
     while (infile)
     {
       outfile << n << endl;
       infile >> n;
     }
     infile.close(); //closing files
   }
  outfile.close();
  return 0;
}
```

```
[lee@bobby cs135sampledir]$ more filedata
10 14 23 18
25
[lee@bobby cs135sampledir]$ g++ samplefile.cpp
[lee@bobby cs135sampledir]$ ./a.out
Please enter name of input file
filedata
[lee@bobby cs135sampledir]$ more myresults
10
14
23
18
25
[lee@bobby cs135sampledir]$ ./a.out
Please enter name of input file
nofile
file not opened for input
```

C++ file streams

Opening a file –

Method 1 for opening a file – use if you know the actual name of the file (source file has to be changed if file name changes)

```
ifstream infile;  //variable that represents input file
ofstream outfile; //variable that represents output file
     //mydatafile is the name of the file in your account
infile.open("mydatafile"); //it is being opened for input
outfile.open("results"); //open a file called results for output
```

Method 2 for opening a file – use if you don't know the actual name of the file and want to allow the user to provide the file name (more general approach)

interactively prompt user for the name of the file

```
ifstream infile; //infile will represent the input file
string fname; //actual name of file
//prompt for name of input file to be opened
cout << "Please enter name of input file" << endl;
cin >> fname;   //read name of file to open

//open input file for reading
infile.open(fname.c_str());
/* c_str() is a function that converts a string variable into a null-
terminated array of characters, which is
the type of parameter the open function expects */
```

Closing a file – when finished reading or writing

```
  infile.close();
  outfile.close();
```

Passing parameters –

- cin and cout are declared in iostream and are global by default
- all other file stream variables must be declared by the programmer
- do **NOT** declare file stream variables on a global level
- file stream variables should be declared locally
- if 2 or more functions need access to the same file stream variable, it **MUST be passed by reference**

C++ file streams

```
//Open an input file called "paylist". Prompt the user for the name
//of the output file. Read employee names, hours worked and pay rate.
//Compute pay due. Write pay information for each employee and total
//due to output file. ~lee/cs135sampledir/payroll.cpp
#include <iostream>
#include <cctype>
#include <string>
#include <fstream>
#include <iomanip>
using namespace std;
double figure_pay(double,double);
void writeinfo(ofstream&,string,string,double);
void format(string&);
int main()
{
  ifstream input;        ofstream output;
  string filename,lname,fname;
  double hrs,rate,pay,totalpay=0.0;
  input.open("paylist");
  cout << "Enter name of output file" << endl;
  cin >> filename;
  output.open(filename.c_str());
  output << fixed << setprecision(2);
  output<<left<<setw(25)<< "Name"<<right<<setw(11) <<"Amt Due\n";
  input >> lname;
  while (input)
    {
      input >> fname >> hrs >> rate;
      pay = figure_pay(hrs,rate);
      writeinfo(output,lname,fname,pay);
      totalpay = pay + totalpay;
      input >> lname;
    }
  input.close();
  output<<left<<setw(25)<<"Total payroll:"<< right << setw(4) <<"$"
        <<setw(7)<<totalpay <<endl;
  output.close();
  return 0;
}
double figure_pay(double hours,double rate)
{
  if (hours <= 40.0)
    return hours * rate;
  else
    return 40.0*rate + 1.5*rate*(hours-40.0);
}
void writeinfo(ofstream& out,string lname,string fname,double pay)
{
  format(lname);  format(fname);
  out<<left<<setw(25)<<lname+','+fname<<right<<setw(4)<<"$"<<setw(7)
     <<pay<<endl;
}
void format(string& word)
{
  word[0] = toupper(word[0]);
  for (int i=1; i<word.length(); i++)
    word[i] = tolower(word[i]);
}
```

```
[lee@bobby ]$  more paylist
smith jack 30 10
JONES mArY 40 12
aDamS biLL 43 8
```

```
[lee@bobby ]$ g++ payroll.cpp
[lee@bobby ]$ ./a.out
Enter name of output file
payresults
[lee@bobby ]$ more payresults
Name                        Amt Due
Smith,Jack                  $ 300.00
Jones,Mary                  $ 480.00
Adams,Bill                  $ 356.00
Total payroll:              $1136.00
```

C++ file streams

```
#include <iostream> //~lee/cs135sampledir/filestream2.cpp
#include <string>
#include <cctype>
#include <fstream>
#include <iomanip>
using namespace std;
int countcaps(string);
int main()
{
  string word;
  int capcount=0;  int wordcount=0,charcount=0;
  string inputfilename, outputfilename;
  ifstream inf;
  ofstream outf;   ofstream statf;
  cout << "Enter name of input file" << endl;
  cin >> inputfilename;
  inf.open(inputfilename.c_str());
  cout << "Enter name of word list file" << endl;
  cin >> outputfilename;
  outf.open(outputfilename.c_str());
  outf << "List of Words" << endl;
  inf >> word;
  while (inf)
    {
      wordcount++;
      charcount = charcount + word.length();
      outf << word << endl;
      capcount = capcount + countcaps(word);
      inf >> word;
    }
  inf.close();
  outf.close();
  statf.open("statsfile");
  statf << fixed << setprecision(3);
  statf<<"# of words found in "<<inputfilename
       << ": "<< wordcount << endl;
  statf<<"# of capital letters: "<<capcount
       << endl;
  statf<<"Ave word length: "
       <<double(charcount)/wordcount
       << endl;
  statf.close();
  return 0;
}
int countcaps(string word)
{
  int count=0;
  int n=word.length();
  for (int i=0; i<n; i++)
    if (isupper(word[i]))
      count++;
  return count;
}
```

```
[lee@bobby]$ more sometext
Here is a short
file. It will be used
to test this
filestream program.
[lee@bobby]$ g++ filestream2.cpp
[lee@bobby]$ ./a.out
Enter name of input file
sometext
Enter name of word list file
wordlist
[lee@bobby]$ more wordlist
List of Words
Here
is
a
short
file.
It
will
be
used
to
test
this
filestream
program.
[lee@bobby]$ more statsfile
# of words found in sometext: 14
# of capital letters: 2
Ave word length: 4.071
```