CSC 215

Math and Computer Science

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

A Legacy of Excellence

Output Files

- Declare a ofstream object

```
      ofstream fout;
```

- *fout* is a general purpose identifier, you can use any name that follows the rules for naming variables
- You will need different identifiers if you are using more than 1 output file at a time.

```
      ofstream fout1, fout2;
```

Opening the output file

- Follow the same technique as an input file.
- If you know the name of a file at declaration

```
ofstream fout ("myfile.txt");

char filename[30] = "myfile.txt";
ofstream outputFile( filename );
```

- Do not know name of file until runtime. (Usual way)

```
char filename[30];
ofstream fout;
cin >> filename;
fout.open( filename );
```

- If the file opens, the contents will be truncated.

Testing That Files Opened

- output file could fail to open for many reasons
  - No space on the drive
  - File is in use by another program
  - File permissions prevent you from accessing
- Two ways to test

```
ofstream fout("somefile.txt");
if( !fout )
    cout << "File somefile.txt did not open" << endl;

or

if( !fout.is_open() )
    cout << "file somefile.txt did not open" << endl;
```

SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

Closing Files

- Always close all files when exiting a program.

```
ofstream fout1, fout2;
:
fout1.close();
fout2.close();
```

- Ensures that operating system keeps the file in a good state
- Ensures that all data is written to file before OS closes it
- Allows other programs to open the file.
- Frees resources – a limited number of files can be opened at a time by any one user in Unix.
- IMPORTANT: if file variable will be reused on another file, follow the fout.close() with a fout.clear() – clears all error flags on file.

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Using Files

- Treat it like monitor output
  - You know what is in the file and the order in which the information is ordered.

```
int num = 9;
double x = 3.14;
char name[100]= "Roger Schrader", word[30]="Programming";

fout << fixed << showpoint << setprecision( 3 );
fout << num << " ";
fout << x << endl;
fout << word << " - ";
fout << name << endl;
```

SOUTH DAKOTA
SCHOOL OF MINES & TECHNOLOGY

Changing output file modes

- Can preserve the contents when file is opened.
  - `ios`::out, `ios`::trunc, `ios`::app, `ios`::ate and others

- Affects the default behavior of opening a file
  - `fout.open( "Somefile.txt", ios::out | ios::trunc); // default, delete data`
  - `fout.open( "Somefile.txt", ios::out | ios::app);   // preserve contents`
  - `Fout.open( "somefile.txt", ios::out | ios::ate );  // preserve contents`

- `ios`::app – open for output and position at end of file. Write all output to end of the file.
- `ios`::ate – open for output and position at end of file. Output can be written anywhere in the file.

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Example 1

- Generate a file to be used by the second input example. Place 10 numbers per line.

```
#include <fstream>
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

int main()
{
      int i;
      int limit;
      int number;
```

SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

Example 1

```
ofstream fout;

srand( unsigned( time(NULL) ) );

fout.open( "numbers.txt" );
if( !fout )
{
     cout << "Unable to open output file: "
          << "numbers.txt" << endl;
     return 1;
}
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Example 1

```
        limit = rand();

        for( i=0; i<limit; i++)
        {
                number = rand();
                fout << number << " ";
                if( (i+1)%10 == 0)
                        fout << endl;
        }
        fout.close();
        return 1;
}
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Clear Member Function

- If any error flags are set, it will unset them.

  Syntax: `fout.clear();`

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Tellp Member function

- Returns an integer that represents how many bytes you are from the beginning of the file.
  - If file was just opened, fout.tellp() returns a 0;
  - If the program has been writing to the file in ios::ate, it might return 100.
  - If the program was done writing when opened with ios::trunc or ios::app, it would return the number of bytes in the file

```
cout << fout.tellp();
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Seekp Member Function

- Allows us to move around in the file.

  Syntax: fout.seekp(int bytesToMove, offset way);
- + bytesToMove moves your position towards the end
- - bytesToMove moves your position towards the beginning
- Way – has three constant value
  - ios::beg – from the beginning of the file
  - Ios::cur – from the current position in the file
  - Ios::end – from the end of the file

SOUTH DAKOTA M SCHOOL OF MINES & TECHNOLOGY

Example 2

```
fout.seekp( 0, ios::end );
cout << fin.tellp() << endl;
```

Example 2

```
fin.seekp( 0, ios::beg );
cout << fin.tellp() << endl;
```

Passing ofstreams

- You can only pass streams by reference.
- Must specifically put & in prototype and definition.
- Anything the function does with the file changes the stream.
- ofstream ADTs do not even allow pass by value

```
void openFiles( ofstream &fout);
```