CSC215

Math and Computer Science

SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

Files

- Three types of files objects
  - ifstream ← used for input
  - ofstream ← used for output
  - fstream ← used for input and output
- Use the proper file for the task.
  - Need to do input, use an ifstream
  - Need to do output, use an ofstream

Declaring Files

- Preprocessor directive needed

```
#include <fstream>
```

- Contain an Abstract Data Type (ADT) to handle input and output from files
- Can include iostream for console input/output as well.
- Iomanip utilities work the same on files as with cin and cout.

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Input files

- Declare a input file stream object

```
ifstream fin;
```

- *fin* is a general purpose identifier, it can be any identifier name
- You can name it anything you wish.
- You can create multiple identifiers for multiple files.

```
ifstream fin1, fin2;
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Opening a file

- Do not know name of file until runtime.

```
char filename[30];
ifstream fin;
cin >> filename;
fin.open( filename );
```

- Provided from command line arguments

```
ifstream fin;
fin.open( argv[1] );
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

The Filename

- Be careful when specifying full path

```
char  filename[30] = "e:\grades\exams.txt";      \\ will not work
```

The '\' is a special character. { \n, \t, \0, \b, \a, \\, \', \"}

```
char  filename[30] = "e:\\grades\\exams.txt";
```

- You could type in "e:\grades\exams.txt" with no problems

```
char  filename[30];
ifstream  fin;
cout << "Enter file name: ";
cin.getline( filename );      \\ could type "e:\grades\exams.txt"
fin.open( filename );
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Testing That Files Opened

- Input file could fail to open for many reasons
  - File does not exist
  - File is in use by another program
  - File permissions prevent you from accessing
- Two ways to test

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

or

```
if( !fin.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.

```
ifstream fin1, fin2;
:
fin1.close();
fin2.close();
```

- Ensures that operating system keeps the file in a good state
- Especially important for output files (later)
- 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 fin.close() with a fin.clear() – clears all error flags on file.

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Using Files

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

```
int num;
double x;
char name[100], word[30];
string str;
```

```
fin >> num;
fin >> x;
fin >> word;
fin.getline( name, 100);
getline( fin, str );
```

Example 1 – read in 100 numbers

```
ifstream fin;

fin.open( argv[1] );

if( !fin.is_open() )
{   cout << "Unable to open file: ";
    exit(0);
}

for( i=0; i<100; i++)
    fin >> data[i];
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Example 2 – Read Until EOF

```
ifstream fin;

fin.open( argv[1] );

if( !fin.is_open() )
{   cout << "Unable to open file: ";
    exit(0);
}

int i=0;
while ( i < ARRAYSIZE && fin >> data[i] )
    i++;
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Clear Member Function

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

  Syntax: fin.clear();

Tellg Member Function

- Returns an integer value representing how many bytes we are from the beginning of the file.
  - If file was just opened, fin.tellg() returns a 0;
  - If the program has been reading from the file, it might return 100.
  - If the program read all numbers from the file, it would return the number of bytes in the file.

Seekg Member Function

- Allows us to move around in the file.

  Syntax: fin.seekg(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
SCHOOL OF MINES
& TECHNOLOGY

Example 2

```
fin.seekg( 0, ios::end );
cout << fin.tellg() << endl;
```

SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

Example 2

```
fin.seekg( 0, ios::beg );
cout << fin.tellg() << endl;
```

Example 3

```
int *ptr;
int size=0, num;
while ( fin >> num )                    ← error flag set
    size++;
ptr = new(nothrow) int [size];
fin.clear( );                           ← clear flags
fin.seekg(0,ios::beg);
int i=0;
while( fin >> ptr[i] )
    i++;
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

ifstream behavior

- Can do anything with your input file streams that you can do with an istream (cin).
- Reading from file uses same functions as reading from keyboard.
  - >>, get, getline
- Has the same member functions
  - .eof(), .ignore(), .clear() …….

Passing ifstreams

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

void openFiles(ifstream &fin);

SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY