

CSC112 Spring 2011
Fundamentals of Computer Science
Summary of Text Files

Text Files

Text files can be thought of as a sequence of ASCII characters. Even if the text file contains "numbers", (e.g., 42,43,44), these numbers are represented by a string of ASCII characters.

Writing Text Files

Files are written in C++ using an object of class **ofstream**. There are several member functions (methods) in this class. The most useful ones are listed below:

- **open** – Open a file before writing it.
- **put** – Write a single character to a file.
- **close** – Close the file when done writing.
- **fail** – Check for failure after opening.

Notice there is no "putline" function. There is no need for one, since the stream operator << may be used with an open file.

To use these functions, we must declare an object of type **ofstream**. For example:

```
ofstream tfile ;
```

Using the **open()** function (for a text file):

```
tfile.open( xxxx ) ;
```



File name

C-style character string

Using the **put()** function:

```
tfile.put( xxxx ) ;
```



Character (data type char) to be written

Input xxxx is not an address

Using the **close()** function:

Easy! There are no parameters. Example: `tfile.close()`

Using the fail() function:

Also easy! There are no parameters. Example: `tfile.fail()`

Using the stream operator: The stream operator may be used with an open text file similar to its use it with `cout`. For example:

```
ofstream tfile ; // Declaration.

tfile.open( "a_text_file" ) ;
tfile << "Hello, this is a line of text output to the file." << endl ;
tfile.close() ;
```

Reading Text Files

Files are read in C++ using an object of class **ifstream**. There are several member functions (methods) in this class. The most useful ones are listed below:

- **open** – Open a file before reading it.
- **get** – Read a single character.
- **getline** – Read a line of input.
- **close** – Close the file when done reading.
- **fail** – Check for failure after opening.
- **eof** – Check for end of file after a read operation.

In addition to these functions, we can also use the stream operator `>>`

To use these functions, we must declare an object of type **ifstream**. For example:

```
ifstream ifile ;
```

Using the open() function: Similar to open for writing.

Using the get() function:

```
ifile.get( xxxx ) ;
```



Character (data type char) to be read

Variable xxxx is passed by reference

example, we consider a text file named `a_text_file` starts with several blanks, then the word “cat”, then a few more blanks, then the word “dog”. The file is shown below:

```
    cat   dog
bird
```

Consider the following program:

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

int main()
{
    ifstream ifs ;
    char word[80] ;

    ifs.open("a_text_file") ;
    ifs >> word ;
    cout << "'" << word << "'" << endl ;
    ifs.close() ;
}
```

When run, the program prints:

```
'cat'
```