Reading & Writing Data Files
Ch 6

1) What are streams, objects, and classes?
2) How can a program read and write a plain text data file?

About Files

- Volatile Storage
  - Data in main memory (RAM) is "volatile":
    It does..
- Non-volatile Storage
  - Computers store data files on "non-volatile storage".
  - Ex: On the..
- Reading/Writing
  - Programs write to files to save/store data.
  - Programs read from files to load data.

File streams

- C++ treats a file as as a stream:
  - An input stream reads from a file: ifstream
    Use..
  - An output stream writes to a file: ofstream
    Use..
- Console Streams
  - cin is an input stream; cout is an output stream
- File Streams
  - Create your own input or output stream which read/write to files.

File streams

- Streams are Objects
  - They store some data, and have functions you can call on them.
  - Call member functions on an object with..
  - Example:

    ```
    if(myStream.fail()) {
         ...handle failure...
    }
    ```

    fail() is a member function: it is a function which belongs to the object myStream.

End of file & fail()

- Often want to read all data in a file
  - Do this by:
    1. Read some data
    2. If reading failed, we’re done
    3. Process data
    4. Goto #1

```
while (true) {
    double value = 0;
    dataFile >> value;
    if (dataFile.fail()) {
        break;
    }
    cout << value << endl;
}
```

Note

Input streams can report when you are at the end of the file:
myFile.eof()

However, handling whitespace at end of file is tricky

More robust to just read until reading fails (as above):
myFile.fail()

Reading a File

- ..

```
ifstream dataFile("data.txt");
if (dataFile.fail()) {
      cout << "Unable to open data file.\n";
      exit(EXIT_FAILURE);
}
```

..

- ..

```
while (true) {
      double value = 0;
      dataFile >> value;
      if (dataFile.fail()) {
            break;
      }
      cout << "Read: " << value << endl;
}
```

..

- ..

```
dataFile.close();
```

Read Ex: Sum numbers in file

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

int main()
{
	// Open the file
	ifstream dataFile("data.txt");
	if (dataFile.fail()) {
		cout << "Failed to open.\n";
		exit(EXIT_FAILURE);
	}

	// Read all values, summing them up.
	double sum = 0;
	while (true) {
		double value = 0;

		dataFile >> value;
		if (dataFile.fail()) {
			break;
		}

		sum += value;
		cout << "Read: " << value << endl;
	}

	dataFile.close();

	cout << "Sum: " << sum << endl;
}
```

... = readText.cpp

Classes and Objects

- Classes and Objects
  - Class: a data type in C++ which allows you to..
  - Objects are..
    - Example with strings:
      string myName;
      - string is a class (the type)
      - myName is an object of type string.
  - When you create variables of type string it.. an object of type string.

Writing a File

- Open the file:

```
ofstream fileOut("data.txt");
if (fileOut.fail()) {
      cout << "Error opening output file.";
      exit(EXIT_FAILURE);
}
```

- Write data:

```
for (int i = 0; i < 10; i++) {
      int value = rand();                         ..
      fileOut << value << endl;
}
```

- Close File:

```
fileOut.close();
```

Write Ex: Write user's values to file

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

int main()
{
    // Open the output file
    ofstream fileOut("data.txt");
    if (fileOut.fail()) {
        cout << "Error opening file.";
        exit(EXIT_FAILURE);
    }

    // Write user values to file
    bool done = false;
    while (!done) {
        double value = 0;
        cout << "Enter a value (-1 to end): ";
        cin >> value;

        if (value == -1) {
            done = true;
        } else {
            fileOut << value << endl;
        }
    }

    // Close the file to flush the output.
    fileOut.close();
}
```

= readText.cpp

Read full lines

- Sometimes program needs whole line of text
  - Ex: csv file (comma separated)
- Can read in a file line-by-line

  Use function..

```
int main()
{
    // Open the file (error checking
    // omitted for space)
    ifstream inputFile("data.txt");

    // Read the file, line by line,
    // and print to screen
    while (true) {
        string nextLine;
        getline(inputFile, nextLine);
        if (inputFile.fail()) {
            break;
        }
        cout << nextLine << endl;
    }
    inputFile.close();
}
```

! readFileLines.cpp

3 ways of leaving: Exit, Return, Break

- exit(EXIT_FAILURE);..
  - Argument:
    0: Success
    1: Failure
  - Defined in cstdlib:
    #include <cstdlib>
- return;
- break;

Summary

- Data files store information between program executions.
- Objects are instances of classes.
  - They have both data and member functions.
- Streams used to read/write files:
  - ifstream: input file stream
  - ofstream: output file stream
  - Use >>, << and getline() to work with files.
- Use exit() to immediately exit the program.