File I/O and STL vectors

21 January 2009
CMPT166
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Pausing at the end of program

- Common issue with console-based programs:
  - When program finishes, it closes the console window right away
- One solution: PAUSE
    - system("PAUSE");
    - Executes the PAUSE.EXE command
- Or use cin to get and throwaway input
- MSVC++ uses this for "Win32 Console" projects

Coding style

- C++ is much more flexible than Python
- Up to you to keep to a consistent coding style
- Common convention: curly braces ({) go right after the function/class name:
  - **int main() {**
  - not
  - **int main()**
  - **{**
- Naming standards: ClassNames, methodNames()
- See online textbook Appendix A for more details

C++ strings

- Standard library; not part of language
  - C++ lang only has arrays of chars
  - But every C++ install has <string>
    - **#include <string>**
    - **void main() {**
      - **string s1, s2;**
      - **s1 = “Hello,”;**
      - **s2 = s1 + “ World”;**
      - **cout << s2 << endl;**
- Declare, initialize, concatenate (+)
- Many more operations; see library reference

File I/O in C++

- I/O is over streams
  - Can be from files, network, other programs...
  - File I/O: #include <fstream>
- Reading from files: creating an ifstream object opens a file for reading
    - **ifstream inputFile("input.txt");**
  - Declare an ifstream object called inputFile and initialize it to open "input.txt"
- Read a line: getline(inputFile, myString);
  - Returns false (0) if end of file

ifstream and ofstream

- Writing to an ofstream is as easy as with cout
- A simple program to copy one line at a time:

```
#include <string>
#include <fstream>
using namespace std;
int main() {
    ifstream in("input.txt");
    ofstream out("output.txt");
    string line;
    while (getline(in, line))
        out << line << endl;
}
```

Standard C++ Library

- Standard C++ Library
  - Not part of C++ language
  - Standard libraries included with every C++ installation
  - Lots of data structures: sets, trees, etc.
  - First one we'll look at: <vector>
- STL (Standard Template Library)
  - Predecessor to Std. C++ Lib.; continues to be developed
  - Many of the same names and ADTs

C++ <vector>

- Like a Python list: flexible size
- Templated: indicate element type in <>
  - **#include <vector>**
  - **vector<string> myParagraph;**
- Can treat like stack: push_back()
  - **myParagraph.push_back(line);**
- Can access like array: .size(), []
  - **for (int i=0; i< myPar.size(); i++)**
    - **myPar[i]**
- More info in STL reference