UNIVERSITY OF MASSACHUSETSS,
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Computer Assignment #0

ECE 670: Advanced System Software Design

October 3, 2018

1 INVERTED INDEXING

This assignment is a warm-up exercise that will refresh your C++ programming memory using the Standard Template Library (STL).

For this assignment, you will write a program in C++ that generates an "inverted index" of all the words in a list of text files. See Wikipedia for more details regarding inverted indexes. The goal of this assignment is to ensure that you are sufficiently up to speed on C++.

1.1 INVERTER INPUT

Your inverter will take exactly one argument: a file that contains a list of filenames. Each filename will appear on a separate line. Each of the files described in the first file will contain text from which you will build your index.
For example:

inputs.txt
—
foo1.txt
foo2.txt

foo1.txt
—
this is a test. cool.

foo2.txt
—
this is also a test.
boring.

1.2 Inverter Output

Your inverter should print all of the words from all of the inputs, in "alphabetical" order, followed by the document numbers in which they appear, in order. For example (note: your program must produce exactly this output):
a: 0 1
also: 1
boring: 1
cool: 0
is: 0 1
test: 0 1
this: 0 1

Alphabetical is defined as the order according to ASCII. So "The" and "the" are separate words, and "The" comes first. Only certain words should be indexed. Words are anything that is made up of only alpha characters, and not numbers, spaces, etc. "Th3e" is two words, "Th" and "e".
Files are incrementally numbered, starting with 0. Only valid, "open-able" files should be included in the count. (is_open() comes in handy here.)
Your program should not produce any other output. In particular, the output should have one space between the colon and the first document number, one space between each subsequent document number, and no spaces at the end of any line.

1.3 Implementation Hints

- Implement the data structure using the C++ Standard Template Library (STL) as a map of sets, as in:
  map<string,set<int>> invertedIndex;
- Use C++ strings and file streams. Sample Code:

  #include<string>
  #include<fstream>

- Make sure that your assignment uses an ifstream, not an fstream. Both ifstreams and fstreams are found in the fstream library.
- Remember, your program needs to be robust to errors in the input file. Files may be empty, the input may contain empty lines, there may be spaces before or after the file-

name on each line of the input file, etc. The only constraint is that each filename will appear on a separate line. Please handle all cases gracefully and with no extra output.

- The *noskipws* operator may be useful in parsing the input:
  input >> noskipws >> c;

1.4 Compiling your code

Use the g++ program to compile your code, and produce a binary called inverter:
g++ inverter.cc -o inverter

To run your code, run:
./inverter input.txt

1.5 Submitting your Code

Please upload one single .cpp file. You should name your file as follows: YourfirstnameYourlastname.cpp. For example my submission would be ArmanPouraghily.cpp.

1.6 Grading

Grading will be based on a small set of test cases and/or a demo.