CECS 429 - Homework 3

1. Given the document "`Sally and Harry watched When Harry Met Sally before they ate popcorn`", how many tokens are in the document? List all of the document types. List all of the terms assuming lowercasing and Porter stemming.
2. From the course textbook, do problem 2.1 (at the end of chapter 2.2). Explain each of your answers with one or two sentences.
3. Suppose you want to index the body of a web page as a document. As discussed in class, you do not want tokens like <html> or other markup tags to appear in the index. Write a paragraph about what text-processing strategies you would use to index an HTML file. Include the following:
   (a) How you would determine which tokens to ignore;
   (b) How you would identify the actual content of the page (the part you want to index);
   (c) How you would extract the title of the document (which might not be the same as the name of the file).
4. Implement parts of a simple inverted index similar to the term-document index you implemented in Homework 1. Download the file `Homework3Files_Java.zip` or `Homework3Files_CSharp.zip` depending on whether you want to complete this lab in Java or C#.
   (a) Create a new project and add all four files from the zip file to the project.
   (b) Spend some time familiarizing yourself with the project as-is. It contains these files:
      i. TokenStream: an interface for classes that can return a sequence of tokens from a file of text.
      ii. SimpleTokenStream: implements the TokenStream interface to provide a stream of tokens with minimal processing, by removing non-alphanumeric characters and lowercasing each token.
      iii. NaiveInvertedIndex: a simple inverted index class. Each string term is mapped to a list of integer document IDs through an appropriate hashtable data structure.
      iv. SimpleEngine: an application that reads a directory of text files and indexes them into the NaiveInvertedIndex, then prints the index.
   (c) Fill in all TO-DO items in NaiveInvertedIndex and SimpleEngine. You do not need to touch TokenStream or SimpleTokenStream.
   (d) Complete main method in SimpleEngine as in Homework 1, so that after indexing all documents, you allow the user to search for single terms and output the set of documents which contain that term.