**CSCE 5200 Information Retrieval and Web Search**

Programming Assignment 1 - Text Parser

**Due date**: March 10th 11:59PM (online submission through Canvas)**Description**:An IR Engine should include at least the following major components: *Text parser*, *Indexer* and *Retrieval* System. Your first programming assignment is to build the first component, the *Text parser* which will be used by subsequent assignments. You can choose your familiar language as the implementation language.

*Note*: If you decide to use C++, you might consider using C++ STL (Standard Template Library), which has all the necessary classes. Get familiar with the different types of containers available in STL along with the methods provided. A Text Parser should include the following functionalities: **Tokenizer:** Reads document into memory, tokenizes to separate words; returns token stream. Basic tokenization rules:

* remove numbers
* ignore if word contains numbers.
* split on all nonalphanumeric characters(such as punctuation marks, spaces,

hyphens, and apostrophes)

* convert to lower case

**WordDictionary:** Build a Dictionary, which assigns each unique word/token to a unique numerical ID and keeps this mapping information (Stemmer Algorithm should be used).

**FileDictionary:** You also need to keep a Dictionary to map each document name to a unique numerical ID.**Data**: We are using the TREC data, which contains multiple documents in a file and tags them separately. So you cannot treat each file as a single document, you need to parse them to separate documents.

**Testing**: You should print out document ids and token streams to see if you properly parse documents. Store the output in a file called "parser\_output.txt" in the following form:

caesar 1

card 2 (token: token ID)

cat 3

……….

FT911-1 1

FT911-2 2 (document name: doc ID)

FT911-3 3

**Document Preprocessing Steps**:-

* Tokenization to handle numbers, punctuation marks, and the case of letters (upper/lower)
* Elimination of stopwords
* Stemming of the remaining words
* Selection of terms for the term dictionary
* Creating the dictionary file (Term Dictionary and Document Dictionary)

**Submit**:

Submit your assignment through Canvas using a .zip file (contain all the files for this assignment, including “parser\_output.txt”, also provide a Readme file for the instruction of how to run your code). Code should be submitted on time, and you may be asked to give a demo.