Assignment Title:	CA2-Mini Google
Module Title:	Applied Data Structures and Algorithms
Weighting:	20%
Deadline:	12th – 30th September

Learning Outcome:

By completing this assignment you will practice and master the following skills:

- ✓ *Working with the fundamentals of Java programming*
- ✓ *Use of object-oriented design principles*
- ✓ *Identifying and fixing defects and common issues in code*

Task:

This assignment requires you to apply the knowledge of Java fundamentals and its advanced concepts to build a simple search engine (or *web crawler*). There are two parts to the search engine: *Crawl Web* and *Search*. Each of these parts also called as "software components" will be described below.

The *crawl web* software component is responsible for fetching web pages from the World Wide Web and adding each web page to a data structure called *Search Index*. You can read more about the Crawler. https://en.wikipedia.org/wiki/Web_crawler

In this project we begin building a *bare bones web crawler*. This means it won't have all the features of an industry scale web crawler but will have enough for us to understand how to program one. All of the crawl web features can be implemented as a set of Java classes which are described below. Consider placing all these Java classes in a package. Packages are not very difficult to understand. Read about how to use packages https://docs.oracle.com/javase/tutorial/java/package/packages.html

The *WebPage* class

The *WebPage* class holds the web page content and it has the following methods:

- getAllLinks() - returns an ArrayList of links in the webpage.
- getWords() - returns an ArrayList of words in the page, after removing the stopwords.
- getContent() - fetches the web page content identified by its URL. Use the example from https://docs.oracle.com/javase/tutorial/networking/urls/readingURL.html to fetch the web page content from the Internet. In this project you can make use of the Java API to directly fetch a real web page from the Internet
- getKeywordFrequency() - given a keyword as input returns the number of times the keyword appears in the web page.

Note: Use this URL https://udacity.github.io/cs101x/index.html to fetch the web page content.

The *WebCrawler* Class

The WebCrawler class must have the following methods:

- A static method *Crawl* with one parameter *seed page*. Seed page is the first page for the web crawler to process. The goal is to start with the seed page, extract all the links from it and repeat

crawling for each link that is extracted, until all the web pages connected to the seed page are fetched. The data type for the seed page can be a String.

The *SearchIndex* Class

- The *SearchIndex* class as the name suggests represents the search index. *SearchIndex* is a data structure that holds keywords with a list of WebPages with at least one occurrence of the keyword in the page. The *Hashtable* class in the Java API is very useful to implement the search index.
  http://docs.oracle.com/javase/7/docs/api/java/util/Hashtable.html
- The *Hashtable* class provides methods to add a keyword and web page to it and to look up for the web pages by providing the keyword as a parameter. The *SearchIndex* should be saved to a *file* when a web page is added to it. On creating the object of the Search Index its data should be retrieved from a file.

Extra Credit: You will award extra credit for additional features if implemented apart from the given specifications.

Assessment Criteria

Grading Criteria	*Points*
Programming Style	5%
Program Logic	20%
Testing	15%
Viva Voce	10%
Total Marks	50%

Please note: The assignment is to be completed as an individual. The submission is to be made via Virtual Learning Environment (*VLE*). Only java source files should be uploaded, no class files. Put all the java files in a zip archive with the following name.

assign1_studentID.zip