**CSCI 1101 – Winter 2017**
**Laboratory No. 9**

***File Input and Output***

*Submission deadline: Saturday, March 25, 2017, 11.55 p.m.*
*The submission requirements and marking scheme are the same as with other labs.*

**Exercise 0: Review the following three exercises that were discussed in the lectures.**

```
//A simple program that illustrates reading from a text file.
//It reads each line from the given text file and displays them on the screen.
import java.util.Scanner;
import java.io.*;
public class FileReadDemo1{
   public static void main(String[] args) throws IOException{
      Scanner keyboard = new Scanner(System.in);
      System.out.print("Enter the name of the file to read from: ");
      String filename= keyboard.nextLine();
      File inFile = new File(filename);
      Scanner input = new Scanner(inFile);
      while(input.hasNext()){
         String line = input.nextLine();
         System.out.println(line);
      }
      input.close();
   }
}
```

```
//This program illustrates how to read from a URL.
//It opens the connection to the specified URL and reads each line and
//displays it. It also counts the total number of characters on the
//webpage.
import java.util.Scanner;
import java.net.URL;
import java.io.*;
public class ReadFileFromURL{
   public static void main(String[] args) throws Exception{
      Scanner keyboard = new Scanner(System.in);
      System.out.print("Enter a URL: ");
      String URLString = keyboard.nextLine();
      URL url = new URL(URLString);
      int count =0;
      Scanner input = new Scanner(url.openStream());
      while (input.hasNext()){
         String line = input.nextLine();
         System.out.println(line);
        count+=line.length();
      }
     System.out.println("The file size is : " + count + " characters");
   }
}
```

```
//A simple file writing program.
//It prompts the user to enter words, one on each line
//and prints them to a file. Note: if the file already exists,
//it overwrites the original file.
public class FileWriteDemo2
{
   public static void main(String[] args) throws IOException
   {
      String name;
      String filename;

      Scanner keyboard = new Scanner(System.in);
      System.out.print("Enter name of the file to write to:");
      filename = keyboard.nextLine();
      PrintWriter outputFile = new PrintWriter(new FileWriter(filename));

      System.out.println("Enter names, end with quit");
      name = keyboard.nextLine();
      while (!name.equals("quit"))
      {
         outputFile.println(name);
         name = keyboard.nextLine();
      }
      outputFile.close();
      System.out.println("Data written to the file");
   }
}
```

**Exercise 1:** Write a program that reads a text file, removes all occurrences of a specified String and stores the remaining words in another text file. You may assume that the text file contains Strings, one on each line. For example, suppose the input text file `lang.txt` contains the following words:

```
Java
C
C++
COBOL
Ada
Pascal
LISP
PERL
Python
Pascal
```

A sample screen dialog is given below:

```
Enter the name of the file to read from: lang.txt
Enter the String to remove: Pascal
Enter the name of the file to write into: lang1.txt
Data written into lang1.txt
```

Now `lang1.txt` will contain the following:

```
Java
C
C++
COBOL
Ada
LISP
PERL
Python
```

**Exercise 2:** Write a program that reads from a URL and searches for a given word in the URL and displays the number of times the word appears. For example, the URL http://liveexample.pearsoncmg.com/data/Lincoln.txt contains the following text.

```
Four score and seven years ago our fathers brought forth on this continent, a new nation,
conceived in Liberty, and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or any nation so
conceived and dedicated, can long endure. We are met on a great battle-field of that war.
We have come to dedicate a portion of that field, as a final resting place for those who here gave their
lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate -- we can not consecrate -- we can not hallow -- this ground.
The brave men, living and dead, who struggled here, have consecrated it, far above our poor power to add or detract.
The world will little note, nor long remember what we say here, but it can never forget what they did here.
It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus
far so nobly advanced. It is rather for us to be here dedicated to the great task remaining before us -- that from these
honored dead we take increased devotion to that cause for which they gave the last full measure of devotion -- that
we here highly resolve that these dead shall not have died in vain -- that this nation, under God, shall have a new
birth of freedom -- and that government of the people, by the people, for the people, shall not perish from the earth.
```

Suppose the word being searched is "great". As you can see from the highlighted portions, there are three occurrences of the word.
The output of your program should say:
`The word great appears 3 times in the URL.`
You may assume that the word being searched is separated by a space (that is, you don't need to search for substrings). You may also assume that the word being searched is case-sensitive (that is, you don't need to search for `Great` if the given word is `great`).

**Exercise 3**: Write a program to create a file named numbers.txt. Create 100 random integers (you can assume some reasonable range for the integers, say, between 1 and 10000) and write them into numbers.txt. Read the data back from numbers.txt and display them in decreasing order.