# Department of Computing

**CS 212: Object Oriented Programming**

**Class: BESE-11AB**

**Lab 12: Files & Streams**

## Lab Manual Upload Date: June 9th, 2021

**Instructor: Ms. Hania Aslam**

### Learning Objectives

The learning objective of this lab is to understand and practice the concept of text file handling in Java.

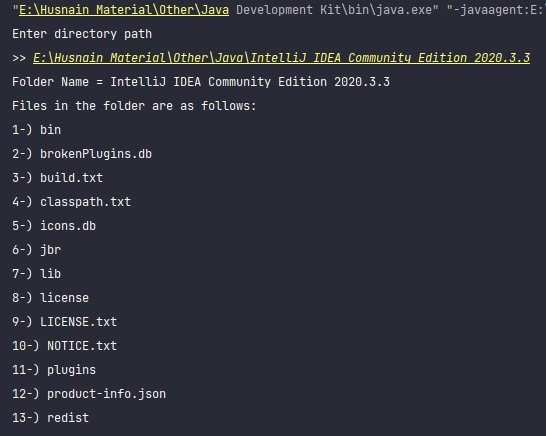
### Lab Task #1

Write a Java program to get a list of all file/directory names inside the Netbeans Folder.

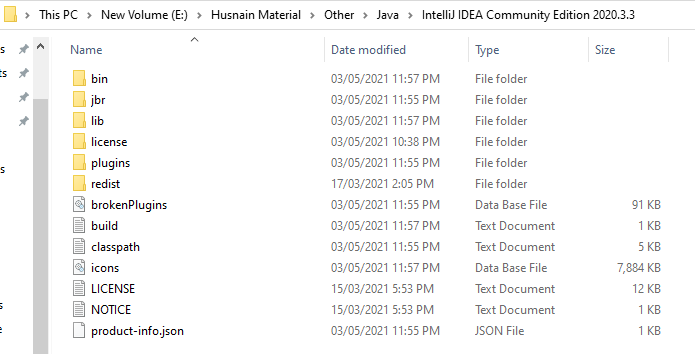
**Code:**

package com.company;  
  
*//Importing relevant classes*import java.io.File;  
import java.util.Scanner;  
import java.io.IOException;  
  
*//Creating the class*public class Task01 {  
 *//Main method* public static void main(String[] *args*) throws IOException{  
  
 *//Creating an instance of Scanner class* Scanner sc = new Scanner(System.*in*);  
 *//Taking the directory path input from the user* System.*out*.println("Enter directory path");  
 System.*out*.print(">> ");  
 String directory = sc.nextLine();  
 *//Creating a file object* File path = new File(directory);  
  
 *//Checking whether the file object is a directory* if (path.isDirectory()){  
 *//Printing folder name and files in the specified folder* System.*out*.println("Folder Name = " + path.getName());  
 System.*out*.println("Files in the folder are as follows:");  
 int sr\_no = 1;  
 *//Looping through the list of files to print all files in the folder* for (String i : path.list()){  
 System.*out*.println(sr\_no + "-) " + i); *//Formatting the output* sr\_no++;  
 }  
 }  
 }  
}

**Output Screenshot:**

****

**Folder Screenshot:**

****

### Lab Task #2

The following paragraph has been taken from an internet source. Store this text in a file and write a program that reads and displays number of total lines and words present in the paragraph:

Automatic number-plate recognition (ANPR) is a technology that uses optical character recognition on images to read vehicle registration plates to create vehicle location data. It can use existing closed-circuit television, road- rule enforcement cameras, or cameras specifically designed for the task. ANPR is used by police forces around the world for law enforcement purposes, including to check if a vehicle is registered or licensed. It is also used for electronic toll collection on pay-per-use roads and as a method of cataloguing the movements of traffic, for example by highways agencies.

Source: <https://en.wikipedia.org/wiki/Automatic_number-plate_recognition>

**Code:**

package com.company;  
  
*//Importing relevant classes*import java.io.\*;  
import java.util.StringTokenizer;  
  
*//Creating class*public class Task02 {  
 *//Main method* public static void main(String[] *args*) throws Exception {  
  
 *//Creating a file object 'f'* File f = new File("textfile2.txt");  
 *//Creating the file called 'textfile2.txt'* f.createNewFile();  
  
 *//Creating objects of FileWriter and BufferedWriter class to write to the file* FileWriter fileWriter = new FileWriter(f);  
 BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);  
 *//Writing to the file* bufferedWriter.write("Automatic number-plate recognition (ANPR) is a technology that uses optical\ncharacter recognition on images to read vehicle registration plates to create\nvehicle location data. It can use existing closed-circuit television, road- rule\nenforcement cameras, or cameras specifically designed for the task. ANPR is\nused by police forces around the world for law enforcement purposes, including\nto check if a vehicle is registered or licensed. It is also used for electronic\ntoll collection on pay-per-use roads and as a method of cataloguing the\nmovements of traffic, for example by highways agencies.");  
 *//Closing both the file after writing* bufferedWriter.close();  
 fileWriter.close();  
  
 *//Creating objects of FileReader and BufferedReader class to read from the file* FileReader filereader = new FileReader(f);  
 BufferedReader bufferedReader = new BufferedReader(filereader);  
 String line;  
 int i;  
 int line\_counter = 0;  
 int word\_counter = 0;  
 *//Reading lines and counting them and words simultaneously* while ((line = bufferedReader.readLine()) != null){  
 System.*out*.println(line);  
 line\_counter++;  
 StringTokenizer st = new StringTokenizer(line);  
 word\_counter += st.countTokens();  
 }  
  
 *//Printing the output* System.*out*.println();  
 System.*out*.println("Number of lines = " + line\_counter);  
 System.*out*.println("Number of words = " + word\_counter);  
 }  
}

**Output Screenshot:**

****

### Lab Task #3

Create a small text file named “Sample.java” (containing 10-12 lines approximately). Write a Java program called ProcessSample.java .Your program should prompt the user for an input ﬁle location (the text ﬁle you created in the ﬁrst step) & should make sure this ﬁle exists on system and is not a directory. It should keep asking the user for input until a valid existing ﬁle is provided by the user.

Next, your program should prompt user for another ﬁle name, for storing output. This ﬁle does not need to exist beforehand and it must be different from the input ﬁle. Read through the input ﬁle, line by line, and count the occurrences of the word “the” in each line. For each line of input, you should write the line in the output ﬁle, followed by the number of occurrences of the term “the”. For example, if the input line is “

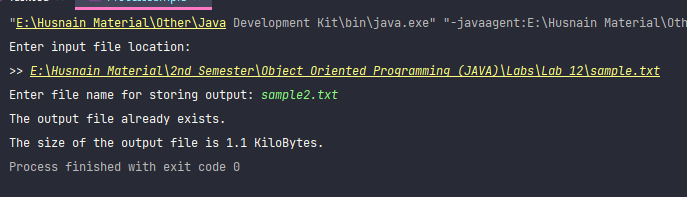
The Magician's Wand is poised with great balance for the

seasoned magician.”, the output line should be “The Magician's Wand is poised with great balance for the seasoned magician. [2]”. Include appropriate error checks on the file arguments, e.g., missing input file, non-readable input file. Once you have finished writing to the output file determine and display its size in kilobytes.

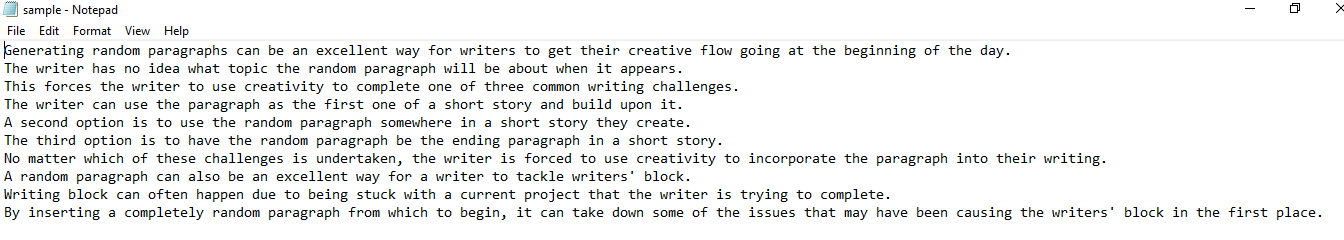
**Code:**

package com.company;  
  
*//Importing relevant classes*import java.io.\*;  
import java.util.Scanner;  
import java.util.StringTokenizer;  
  
*//Creating our class*public class ProcessSample {  
  
 *//Creating method for calculating the occurrence of a specific word in a line* public int getFrequency(String *line*, String *word*){  
 StringTokenizer st = new StringTokenizer(*line*);  
 int tokens = st.countTokens();  
 int frequency = 0;  
  
 for (int i = 0; i < tokens; i++){  
 if (*word*.equals(st.nextToken().toLowerCase())){  
 frequency++;  
 }  
 }  
 return frequency;  
 }  
  
 *//Creating a method for getting the file size* public double getSize(File *f*){  
 return (double) *f*.length()/1000;  
 }  
   
 *//Main method* public static void main(String[] *args*) throws IOException {  
 Scanner sc = new Scanner(System.*in*);  
 String input\_location;  
 File input\_file = null;  
 *//Do while loop for the correct input file location by the user* do{  
 System.*out*.print("Enter input file location: \n>> ");  
 input\_location = sc.nextLine();  
 input\_file = new File(input\_location);  
 }while(!input\_file.exists() || input\_file.isDirectory());  
  
 *//Creating output file for storing output data* String output\_location;  
 System.*out*.print("Enter file name for storing output: ");  
 output\_location = sc.nextLine();  
 File output\_file = new File(output\_location);  
 if (output\_file.exists()){  
 System.*out*.println("The output file already exists.");  
 }else{  
 output\_file.createNewFile();  
 System.*out*.println("The file was created.");  
 }  
  
 *//Creating the object of our class to refer to its methods* ProcessSample ps = new ProcessSample();  
 *//Opening the streams for reading and writing* FileReader fileReader = new FileReader(input\_file);  
 BufferedReader bufferedReader = new BufferedReader(fileReader);  
 FileWriter fileWriter = new FileWriter("sample2.txt");  
 BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);  
 String line;  
 *//Writing to the specified output file* while ((line = bufferedReader.readLine()) != null){  
 bufferedWriter.write(line + " ");  
 bufferedWriter.write("[" + ps.getFrequency(line,"the") + "]");  
 bufferedWriter.write(System.*lineSeparator*());  
 }  
 *//Closing all streams after use* bufferedWriter.close();  
 fileWriter.close();  
 bufferedReader.close();  
 fileReader.close();  
  
 *//Printing the file size of output file.* System.*out*.printf("The size of the output file is %.1f KiloBytes.",ps.getSize(output\_file));  
   
 }  
}

**Output Screenshot:**

****

**Sample.txt (Reading File):**



**Sample2.txt (Output File):**

### 

### Hand in

Hand in the source code from this lab at the appropriate location on the LMS system. You should hand in a single compressed file named LAB\_12\_<Your CMS\_ID. Your\_NAME >.docx.

### To Receive Credit

1. Comment your program heavily. Intelligent comments and a clean, readable formatting of your code account for 20% of your grade.
2. The lab time is not intended as free time for working on other assignments.

