

CSE 5345/7345

Assignment 2. (To-Be-Graded) Due: Oct 8 [Oct 11 distance]

**A. Create a NetBeans project called A2Counter.**

Within the package, create a package called a2package. All the classes for this assignment should be defined in this package.

**B. In the a2package, write a java class: A2Interface.java.**

```
package a2package;

public interface A2Interface {
    public void addWord(String x) throws NotWordException;
    public void addMultiWords(String commaDelimString) throws
        BadCommaDelimStringException;
    public String getUniqueWordsSorted();
    public int getCount(String word);
    public int getTotalWords();

    void reset();
}
```

C. To compile t A2Interface.java, you will need to define within the package, your own Exception classes: NotWordException and BadCommaDelimStringException

**D. When you get the interface to compile, write an implementation called A2.java. This source file should contain a javadoc comment of the form:**

```
/** This class displays the values of three parameters:
 * @author Cameron, Sasha
 */
```

**The method descriptions are as follows:**

```
void addWord(String s) throws NotWordException;
    track the word so you can return the number of times it has been added; use
    regexp to check the string; throw NotWordException if the incoming string contains
    anything other than upper/lower case alpha characters, or contains zero characters.

public void addMultiWords(String commaDelimString) throws
BadCommaDelimStringException;
    incoming string may contain one or more alpha words separated by commas; track
    all words; use regexp to check string and throw BadCommaDelimStringException if the
    string contains other than alpha words and commas.

public String getUniqueWordsSorted();
    return all the words, sorted alphabetically, separated by commas.

public int getCount(String word);
    return the count of how many times a given word has been added

public int getTotalWords();
    return the number of unique words you are tracking

void reset();
    resets your object to scratch - just as when your program first started.
```

E. With your A2.java open in netbeans, use the Tools..Create Junit tests option. This will Create a junit test class called testA2.java that tests the correctness of your A2.java. You can run the test under the RUN menu .

The default is a test for every method. Initially the code all fails. You will need to modify these methods and add additional methods to run additional tests. The creative part of this exercise is to come up with tests that will catch errors in the code.

Your A2Test.java class will also be used to test an implementation of the A2Interface that contains several errors. Your test cases should be written to uncover errors the poorly written implementation.

F. In NetBeans, select Run.. Clean and Build - which will create a jar file called A2Counter.jar in your project dist directory.

SUBMIT:

Create a zip file, named A2yourid.zip that contains:

- A2Counter.jar (at the top level of the zip file.)
- your src folder from your project folder.
- your test folder from your project folder (this will contain your A2Test.java code)

Check your files before submitting:

- is your package named according to the spec?
- do you have a javadoc comment in **all** your source files?
- Make certain your jar file contains all your class files. (A jar file is just a zip, so you can open a jar file with WinZip). Source files not required in the jar file.