

COMP1006/1406 – Summer 2016

Submit a single file called `assignment4.zip` to cuLearn.
There are 100 marks possible marks. The assignment it out of 100.

1: Balanced Parentheses [25 marks]

A stack is a data structure where data is accessed using the LIFO (last in first out) approach. In this problem, you will use a stack to check whether a string has balanced parentheses “()” (and possibly brackets “{}”, “[]”), or not.

For this problem, you can receive up to 80% by testing for balanced parentheses. The remaining 20% will be for a more general solution that checks for all parentheses and balanced brackets (curly and square).

A string that has balanced parentheses and brackets will be said to be **balanced**. Any character that is NOT one of ‘(’, ‘)’, ‘{’, ‘}’, ‘[’, or ‘]’ are not important when deciding if a string is balanced and can be ignored.

We will define balanced as follows. A string `str` is balanced

- if `str` does not contain a parenthesis or bracket symbol, or
- if `str` consists of a balanced string surrounded by opening and closing parentheses or matching brackets. That is, `str` is “(b)”, or `str` is “[b]” or `str` is “{b}”, where `b` is any balanced string, or
- if `str` is the concatenation of any two balanced strings. That is, `str` is “bc”, where `b` and `c` are any balanced strings.

➡ Write a Java class called `Balanced` that has two static methods `isBalanced(String)` and `numberOfBalancedStrings(String[])`. Use the provided template and write the body of the method according to the specifications provided.

Your `isBalanced()` method must use the `java.util.Stack` class (in a way that solves the problem) to receive any grades for this problem.

(<http://docs.oracle.com/javase/8/docs/api/java/util/Stack.html>)

Mark breakdown: 20 marks for balanced parentheses, 5 marks for general solution

Put your `Balanced.java` file in your `assignment4.zip` file.

Examples: The following strings have balanced parentheses

- “”, “()”, “()()”
- “cat”, “c(at)”, “(hello)(kitty)”
- “if((x-y) < 4) || (x > 12){”

- “()((s))()()()()x()(y)(x).....()(ccccc(w))ssss()”

The following strings do NOT have balanced parentheses

- “)”, “)(a)”, “)a(”

The following strings have balanced parentheses and brackets

- “”, “()”, “[]”, “{}”, “{[]”, “[()]”, “({[])”, “() [] {}”, “([]) [{}] []”
- “for(int i=0; i<12; i+=1){x[i]+=1;}”

The following strings do NOT have balanced parentheses and brackets

- “[”, “{”, “}”, “]”, “(”
- “for(int i=0; i<12; i+=1){”

2: Top 10 [25 marks]

In this problem, you will complete a program that reads in a text file and outputs the top ten words that appear most frequently in the file. Reading data from the file is done for you in the provided skeleton file.

► Complete the `WordStats` program that is provided. Your program will take a single command line argument that is the name of a text file (e.g., “file.txt”) in the same directory as your program. The program will then read every “word” in the text file and output a graph of the relative frequency of the 10 most frequent words in the file. A word is a maximal sequence of characters without whitespace. The output of the program should look something like the following:

Total number of words is 1243

```
-----
the      |##### [851]
cat      |##### [425]
an       |##### [409]
curious |##### [408]
android |##### [84]
pizza   |### [52]
dog      |## [33]
eel      |## [32]
end      |## [32]
what     |# [17]
-----
```

finished in 1.343 seconds

A count of all words is given and then a plot that shows each of the words and their relative frequency. Each word is given along with a graphical representation of the relative frequency of that word. The most frequent word should have fifty # symbols and the others based on this scaling. Use simple rounding to determine number of # symbols. The actual

count of each word should also be displayed as shown above (in square brackets). For example, the word “the” was most frequent appearing 851 times. The time message at the end is provided in the skeleton code.

Be sure to only modify `main` in the section indicated. You may add any helper methods you need.

Your plot may contain less than 10 words (if the text file has less than 10 unique words) and it may contain more than 10 words (if there is a tie and there are more than 10 words with the greatest frequencies).

You must use a `Hashtable` to keep track of the frequency of each word in the file.

(<http://docs.oracle.com/javase/8/docs/api/java/util/Hashtable.html>)

When considering “words” in the file, you should be ignoring punctuation (period, comma, question mark, exclamation mark, quotation marks) but keep apostrophes and hyphens. For example, “that’s” should be considered a single word.

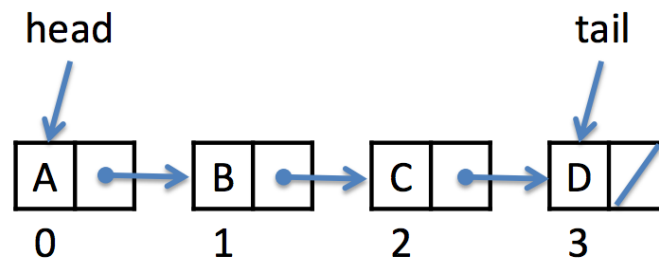
In this problem you will initially need to use a hashtable but then move the data to another structure to complete the problem.

Mark breakdown: 25 marks for correctness

Put your `WordStats.java` file in your `assignment4.zip` file.

3: Linked List [25 marks]

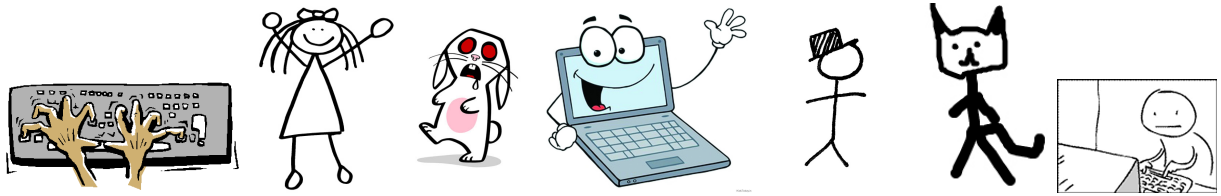
In this problem you will implement a linked list class for Strings.



➡ Write a `LinkedList` class that extends the provided `ALinkedList` abstract class. The abstract class provides methods to add and remove a string from the front or back of the list. You will need to implement two additional methods: `insert` and `extract`. The specifications are found in the `ALinkedList.java` file.

Mark breakdown: 25 marks for correctness

Put your `LinkedList.java` file in your `assignment4.zip` file.

4: Drawing [25 marks]

Think about your experience so far in COMP1006/1406. Think about what you have learned and what you have done. Your task in this problem is to draw a picture that expresses this reflection.

My hope in asking you to draw a picture is that you will critically reflect on the learning objectives you have achieved in this course so far. It should also make this assignment a bit lighter than the others. The intention is that this problem should not cause you stress. Do not worry about your “artistic ability”. You will not be graded on that at all. Have fun.

Your submission should be in PDF format (if you prefer to hand in a physical copy then contact me and we will make arrangements for this). Ideally, the size of your drawing should be a standard letter size in horizontal orientation. If you submit your picture in a file called [Public.pdf](#) then you are agreeing to have your picture shown to the class. If you prefer that your picture not be shown then submit it in a file called [Private.pdf](#).

Offensive/rude/insensitive submissions will receive zero marks and may be forwarded to the Dean’s office depending on the severity. (This has never happened before.)

Mark breakdown: 25 marks for submission

Put your [Public.pdf](#) or [private.pdf](#) file in your `assignment4.zip` file.
Or submit your hardcopy to me.

Submission Recap

A complete assignment will consist of a single file (`assignment4.zip`) with the four files [Balanced.java](#), [WordStats.java](#), [LinkedList.java](#), and either [Public.pdf](#) or [Private.pdf](#).