

University of Newcastle
Discipline of Computing and Information Technology
Semester 1, 2020 - SENG1120/6120

Assignment 1

Due using the Blackboard Assignment submission facility:

11:59PM – April 3rd, 2020

NOTE: *The important information about submission and code specifics at the end of this assignment specification.*

INTRODUCTION

You are required to build the infrastructure to manipulate data related to text. Your client further specifies that you are to create a class named `LinkedList` to store individual tokens. The `LinkedList` will store each token in a `Node` of the list, using the class `string`.

ASSIGNMENT TASK

You are required to use a **doubly-linked list**, as discussed in lectures, to create your own implementation of the `LinkedList` class. It will use instances of `Node` to store instances of `value_type` (in this assignment, each `Node` will be used to store an instance of `string`). Your list will be used to store the individual words of a sentence, broken up on the spaces; so each `Node` will store a single word. For instance, "The Black Cat" will add three separate words ("The", "Black", "Cat") to the list.

The `LinkedList` class will be used by a main program, to be supplied to you, as well as a `makefile`. You will need to design `LinkedList` and `Node` in a way that it communicates seamlessly with the main program and compiles with the `makefile` also supplied. Please refer to the lecture slides and recordings for guidance on how to implement both classes.

For students in SENG6120, there is an extra requirement:

- (3.0 marks) Extend the member method `void remove(string)` inside `LinkedList`. The extended method pattern matches and then removes all occurrences of an input sentence. For instance, if the linked list contains the sentence "The black cat was sitting on the black mat that was on the black floor", and the input string for `remove(string)` is "on the black", then the resulting linked list must be "The black cat was sitting mat that was floor".

For SENG1120 students who want to be challenged more, the above requirement becomes a bonus question, also worth 3.0 marks; however you can still only score a MAXIMUM of 15.0/15.0.

SUBMISSION

Make sure your code works with the files supplied, and DO NOT change them. For marking, we will add the **Demo and makefile** to the project and compile everything **using this makefile**, together with your own files. If it does not compile or run, your mark will be zero.

Your submission should be made using the Assignments section of the course Blackboard site. **Incorrectly submitted assignments will not be marked.** You should provide the .h and .cpp files related to the linked list and node classes, only, plus an assessment item coversheet. Also, if necessary, provide a `readme.txt` file containing instructions for the marker. Each program file should have a proper header section including your name, course and student number, and your code should be properly documented.

Remember that your code should compile and run correctly using Cygwin. There should be no segmentation faults or memory leaks during or after the execution of the program.

Compress all your files into a *single .zip file*, using your student number as the filename, and appending **'_remove'** if you have attempted the bonus section. For example, if your student number is **c9876543** and you have implemented `remove()` (*including 6120 students*), you would name your submission:

c9876543_remove.zip

The same student who has NOT attempted the `remove()` task would simply name their submission:

c9876543.zip

Submit by clicking in the link that will be created in the Assignments section on Blackboard.

Late submissions are subject to the rules specified in the Course Outline. Finally, a completed Assignment Cover Sheet should accompany your submission.

This assignment is worth 15 marks of your final result for the course (including bonus marks).

Here is a screenshot of the expected output (with the bonus question included).

```
Alexandre@ces249-339952s ~
$ ./assignment1.exe
Start lists:
List 1: The black cat was sitting on the black mat that was on the black floor
List 2: The dog scared the cat and the cat ran away

Concatenating the two lists onto list '1':
List 1: The black cat was sitting on the black mat that was on the black floor The dog scared the cat and the cat ran away
List 2: The dog scared the cat and the cat ran away

Removing the word 'was' from list '1':
List 1: The black cat sitting on the black mat that on the black floor The dog scared the cat and the cat ran away
List 2: The dog scared the cat and the cat ran away

Removing the word 'away' from list '2':
List 1: The black cat sitting on the black mat that on the black floor The dog scared the cat and the cat ran away
List 2: The dog scared the cat and the cat ran

Removing the word 'cat' from both lists:
List 1: The black sitting on the black mat that on the black floor The dog scared the and the ran away
List 2: The dog scared the and the ran

Number of occurrences of 'black' in list 1: 3

Removing 'on the black' from both lists:
List 1: The black sitting mat that floor The dog scared the and the ran away
List 2: The dog scared the and the ran

The program has finished.
Alexandre@ces249-339952s ~
$
```

```
Alexandre@ces249-339952s ~
$ ./assignment1.exe
Start lists:
List 1: The black cat was sitting on the black mat that was on the black floor
List 2: The dog scared the cat and the cat ran away

Concatenating the two lists onto list '1':
List 1: The black cat was sitting on the black mat that was on the black floor The dog scared the cat
and the cat ran away
List 2: The dog scared the cat and the cat ran away

Removing the word 'was' from list '1':
List 1: The black cat sitting on the black mat that on the black floor The dog scared the cat and the
cat ran away
List 2: The dog scared the cat and the cat ran away

Removing the word 'away' from list '2':
List 1: The black cat sitting on the black mat that on the black floor The dog scared the cat and the
cat ran away
List 2: The dog scared the cat and the cat ran

Removing the word 'cat' from both lists:
List 1: The black sitting on the black mat that on the black floor The dog scared the and the ran away
List 2: The dog scared the and the ran

Number of occurrences of 'black' in list 1: 3

Removing 'on the black' from both lists:
List 1: The black sitting mat that floor The dog scared the and the ran away
List 2: The dog scared the and the ran

The program has finished.
Alexandre@ces249-339952s ~
$
```

Dan and Alex

v1.0 (2020-03-11)

v1.01 (2019-03-14) – updated *SUBMISSION para 1* to remove typo – changes highlighted