University of Newcastle
Discipline of Computing and Information Technology
Semester 1, 2018 - SENG1120/6120

Assignment 1

Due using the Blackboard Assignment submission facility:
11:59PM – April 20th, 2018

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 text as in a word processor. Your client further specifies that you are to create a class named `LinkedList` to store text inputted by the user. The `LinkedList` will store each word in a `Node` of the list.

ASSIGNMENT TASK

You are required to use a 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 a single word using a `string` type).

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 provided, and compiles with the `makefile` also supplied. Please refer to the lecture slides and recordings for guidance on how to implement both classes.

SENG6120 students should implement another method, called `reverse()`, that will reverse the contents of the linked list. Please note that you cannot copy the contents of the nodes around. You will have to change the structure of the linked list itself. **SENG1120 students who implement this correctly** will get a 1.5-marks bonus.

SUBMISSION

Make sure your code works with the files supplied, and DO NOT change them. For marking, we will add the main file to the project and compile it using the 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 and submit it in by clicking in a 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.

Compiling and running your files together with the demo file provided should output the following result: