Reign

Walker

4144021

Practical Assessment 2

CSC211 2022

# Linked list report

# 1. Introduction and Background

The problem I was presented was to be able to create a program that contained data on which was captured and stored using linked lists. The program was a register-based program that allowed data to be entered, deleted, and displayed to the user. The data that the user was required to enter is their ID number, name, surname, cell phone number, and information regarding their registration. The entire program was to be created using object orientated programming objects and algorithm-based techniques that allowed for the program to run efficiently and effectively

# 2. Implementation

The program is based on 3 class files including, the Citizen class, CitizenRegister class, and the Node class. The Citizen class is used to create a citizen object, which contains all the user data. The CitizenRegister class is used to implement the singular linked list, that uses the Node class, to create an effective way to store every Citizen object. The program is optimized by using an effective way of storing data and using a single linked list instead of an array. The program is optimised as well by reusing methods such as the is Empty() method, which check whether the program has any data inputted or not, and is reused in multiple method to save on compile time.

# 3. Testing

The challenges presented to me was to have a good understanding of nodes, and how singular linked lists are to be implanted correctly. In the beginning this proved to be my biggest challenge, and I sought out a better understanding of this concept consulting a book, Data Structures and Algorithms in Java by Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser. This helped me tremendously in being able to allow the user to input data, and store that data in cohesive system that allowed for the data to be manipulated by the user effectively, and efficiently. How I tested my program for correctness, was testing every method that is provided to the user. This allowed me to start of by debugging the less difficult methods such as addCitizenAtHead(), which adds data at the begging of the linked list. This allowed me to further understand linked lists and gave me a better understanding to debug much harder methods, such as removeCitizen(), that could potentially be removing a node in the middle of a linked list. This progression of testing lesser problems to much harder ones gave me insight on a huge underlying issue a lot of programmers need to always keep in mind and that’s how the user will interact with their program and consider all the possible inputs and misinputs by the user. The program thus handles many inputs from the user, always provides an output to keep the program running. However, the program does not consider whether the user enters the correct data type when inputting data and thus the program will shut down. The program runs as intended, with the condition that the user enters the correct data type when inserting information.

# 4. Conclusions

The program works as intended and considers and handles a lot of possible inputs from the user. However, the program does not take into account human error and thus any misinput regarding data type will result in the program terminating. This program was solely done by me, and only consulted help from a java textbook to better my understanding of singular linked lists.

# 5. References

Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2014). *Data Structures and Algorithms in JavaTM* (6th ed.). Wiley.

# Plagiarism

I am aware and understand that plagiarism is wrong and punishable according to the UWC policy. I confirm that I have not plagiarized my work neither have I allowed anyone to copy my code or report.

Student Number: 4144021

Student first name and surname: Reign Walker

Date: 2022/03/06