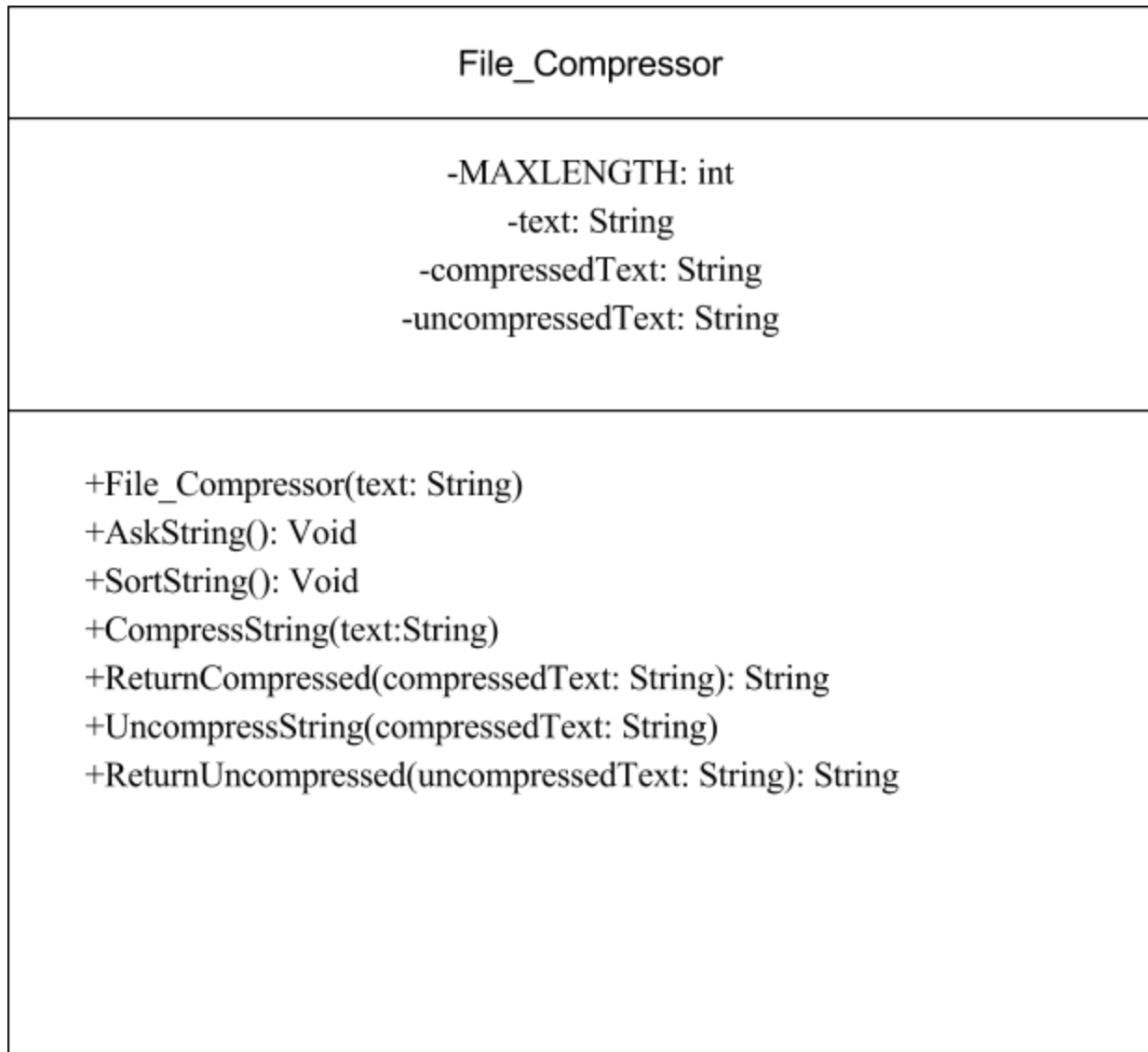


Campbell Lang
CMPT 220
Professor Arias
Project 2 Milestone Report

Abstract: So far, I have worked on a Java program that will try and resolve the issue pertaining to memory. Work that has been completed in the past, as well as items learned in Software Development 1 have culminated in the current project.

Introduction: File Size and memory output has always been an issue. Optimizing seems to be the driving force of computer companies in order to make up for the ever-increasing demand of faster computing capabilities. My program aims to minimize the memory allocated needed for text files. By storing the text in a smaller size, less memory is needed and can be used for more important functions. This paper will help flesh out the details of the program, as well as provide a UML diagram. In addition, this paper will go more in depth into the problem that the program is addressing and show what other work has been done in the realm of decreasing the amount of memory for text functions. At the end, a manual will be provided to show how the program should be used.

Detailed System Analysis: this system is programmed to start with user input initiated by AskString(). The user will enter in text, limited to 50 characters. Afterwards, the text will be converted into an Array and sorted alphabetically by SortString(). Ultimately, the Array will be put through more methods that will compress the code [CompressString()] and return it, [ReturnCompressed()]. Then the method [UncompressString()] will put the compressed text back to normal and finish with returning it in [ReturnUncompressed()].



Requirements: As I said before, file sizes are a major issue in the world we live in. Sending information at a moments notice is how business' stay alive, and the world is kept up to date. Sadly, as everything gets more complicated, the amount of memory needed increases. Therefore, sending files takes longer, as the machine is put under more stress. This system aims to lower the memory needed to send/store text. By compressing the amount of text down into a certain amount of characters then it allows the system to allocate less memory and focus more on

more pressing functions. In the end, the system will return out both the compressed version. If the user wishes, then the system will return back the uncompressed version of the text.

Literature Survey: The most popular application that a “file compressor” by using java code is creating a program that turns files into ZIP files. This achieves the same goal of saving memory, unless needed, and is efficient in doing so. Other than converting files into ZIP files, no other work has been done in the area of file compressors.

User Manual: The user should follow the prompt upon starting the program. It shows that the text they enter has a character limit. Upon entering the text, they will be returned with the compressed version of the text. They will then be prompted if they wish to uncompress the text. If no, then the program is terminated. If they respond yes then the uncompressed version of the text will be returned.