Executive Summary

Part of your completed assignment submission should be an executive summary containing an "Assignment overview" (1 paragraph, up to about 250 words) explaining what you understand to be the purpose and scope of the assignment and a "technical impression" (1–2 paragraphs, about 200–500 words) describing your experiences while carrying out the assignment. The assignment overview shows how well you understand the assignment; the technical impression section helps to determine what parts of the assignment need clarification, improvement, etc., for the future.

During this assignment I learned the fundamental differences between UDP and TCP protocols. Doing this assignment displayed the practical differences between TCP and UDP, such as TCP needing to be connected to a client and server directly prior to sending messages; while as, UDP does not need to be connected and can send messages without a connection. UDP would be closer to a postal service, while a TCP connection is similar to a telephone line. In a postal service a mailer can send their message wrapped up in an envelope (datagram packet) without a response as long as they know the address (IP address) and zip code (port) of the recipient. However, a phone connection needs the other party to pick up prior to any messages being sent through the phone (prior to the answering machine invention). I also gained a better understanding of datagram packet usage and sockets, alongside their purposes.

When doing the assignment I found the TCP the easiest portion to complete and understand, as it was easiest to troubleshoot and conceptually get. There wasn't much technical trouble during this besides the Logging portion, but that was mainly due to a lack of knowledge on how logging worked, or how to print it out into a log/text file. Meanwhile, the UDP issue was much more challenging to work with due to the datagram packets, differing port usages, and other weird miscellaneous issues. The first issue that arose was a weird datagram packet issue where when attempting to send a message containing a word from the hashmap, alongside the system time formatted, and a string value such as "completed." The packet would not send, and when debugging there was an issue stating the message was too long, even though it would run if you rearranged the words in a different order. That issue is still unresolved, but the workaround was to just put the time in the log instead of sending it via the packet. When creating the UDP there was a lack of understanding on my end on how the sender needs to have a different port than the server, so that issue was resolved quickly. However, the last issue is still unresolved where the key-mapping via the UDP does not work properly, while the TCP does. When trying to put over an already stored key it won't overwrite it, it'll only overwrite to the point that matches its characters. For example, if happy:birthday is a key-pair, and i want to put in happy:five it'll store it as happy:fiveday. The code for TCP vs UDP for the key-pair is pretty much the same, so it's unclear why this issue is arising. The last issue was trying to figure out how to get the program to run straight from the command line like "java server.app <IP> <port>." Instead I had to use the scanner function and prompt the user to ask for input.