# Evaluation

The stock calculator application coursework is an assignment that requires the use of functional programming to create solutions for the given tasks in Scala. This evaluation will focus on presenting the ways that functional thinking and programming was applied throughout the application.

Firstly, functional thinking can be described as a programming approach that focuses on values to functions as opposed to imperative programming where the aim is to declare and shape variables. Functional programming also aims to mostly use functions that have no side effect, in other words they have no external impact other than returning the necessary value.

Text

Description automatically generated

In the Stock Calculator application, the “*readFile*” function splits the line of data into the head (first value) and the tail, which includes all values except the first (head). The idea of head and tail in functional programming is one of the core concepts that can be implemented. Instead of using a generic loop, the function uses head and tail to create a map of keys and values.

Another example of functional programming is the use of *Recursion*. A function can be described as recursive if it makes a call to itself, that is when it is called inside its own scope.

Text

Description automatically generated

In the “*find Median*” function, the function contains a definition for another function which then loops through the key and values contained in the map data. Most of the calculation is done first in the if statement block, which is followed by the else statement to evaluate remaining cases. The inner function is then called within the “findMedian” block.

Text

Description automatically generated

In the “getAverage” function, the solution makes use of the functional programming operation called “folding”. A fold can be defined as a way of “getting a single value by operating on a list of values.” (tcl-lang, 2020)

The folding technique used in the application code is used to change a list into the average of the sum of all of its values. This is then converted into a double to return the final value from the application.

The implementation of functional programming in this solution provided several benefits in the process of creating the application. One of the main benefits of this approach was the improved readability and reduced size of code needed each function. Another benefit is that as this approach promotes the use of pure functions, the application of this makes the code easier to understand and their function signatures lets the reader know the exact input and output of each function. The code overall becomes more concise and less cluttered after the implementation of functional thinking & programming approach.

In my opinion, the best way to approach this assignment is to use Scala, because it provides the many benefits mentioned that are associated with functional programming. My second choice of programming language would be Java, as it similar due to the fact that Scala is based on Java. With Java I would use Java FXML to create a graphical user interface that could further enhance the Stock calculator application. All in all, I find that Scala is the best choice for this assignment as it uses the functional programming approach, whereas Java uses imperative programming. However, Java seems to be more supported and popular in the community, so it seems very viable when it comes to programming in my own preference.