Insert dissertation title here, using manual line breaks if necessary

by

Some person

In partial fulfilment of the
requirements for the degree of
MSc
in
Software Development

University of
Strathclyde
Glasgow

Department of
Computer and Information Sciences

August, 2021

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Declaration

This dissertation is submitted in part fulfilment of the requirements for the degree of MSc in Software Development of the University of Strathclyde.

I declare that this dissertation embodies the results of my own work and that it has been composed by myself.

Following normal academic conventions, I have made due acknowledgement to the work of others.

I declare that I have sought, and received, ethics approval via the Departmental Ethics Committee as appropriate to my research.

I give permission to the University of Strathclyde, Department of Computer and Information Sciences, to provide copies of the dissertation, at cost, to those who may in the future request a copy of the dissertation for private study or research.

I give permission to the University of Strathclyde, Department of Computer and Information Sciences, to place a copy of the dissertation in a publicly available archive.

(please tick) Yes [X] No [ ]

I declare that the word count for this dissertation (excluding title page, declaration, abstract, acknowledgements, table of contents, list of illustrations, references and appendices is XXXX.

I confirm that I wish this to be assessed as a Type 1 2 [✓ 3] 4 5 Dissertation (please circle).

Signature:

Date:

Acknowledgements

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Contents

1 Literature Review

The imperative programming paradigm *Overview of the four main programming paradigms* relies on the programmer to instruct the machine to manipulate and manage memory in a particular order and way to perform actions. This is achieved either by creating groups of instructions as proceduresIshida, Sasaki, and Fukuhara, Ishida1991 or by creating objectsLewis, Loftus, and Smith, Lewis 2009, which groups state along with the instructions. Either way, imperative programming relies heavily on the states, also known as variables, which are heavily modified, to create logic.Harrison, Harrison 1997. We going to borrow an example from Harrison's *Introduction to Functional Programming to better illustrate this*.

" Before execution, the state has some initial value $\delta 0$, representing the inputs to the program, and when the program has finished, the state has a new value $\delta' includingtheresult(s).Moreover, duringexecution, eachcommandchangesthestate, whichhasthere f$

$\delta = \delta_0 \rightarrow \delta_1 \rightarrow \delta_2 \rightarrow \ldots \rightarrow \delta_n \rightarrow \delta'$

For example in a sorting program, the state initially includes an array of values, and when the program has finished, the state has been modified in such a way that these values are sorted, while the intermediate states represent progress towards this goal.

"

This allows programmers to have a great amount of control as well as rely on them to ensure the states and functions are operating well. However, this amount of low-level control isn't always desirable as it brings a high likelihood of errors and faults along with them.Ko and Myers, Ko 2005. The use of special linting-type cases allows us to limit the possibility of errors but fundamentally the paradigm relies on human accuracy and precision to create a strong program with a rigid core.

The declarative programming paradigm *Overview of the four main programming paradigms* is a departure from the original imperative programming. Rather than allowing programmers instruct how the states should be modified and changing the states gradually to achieve the desired result, it relies on the compiler rather than the programmer to create instructions. The programmer instead has to *describe* qualities of the desired result, and not how to achieve it. The compiler will understand the requested result and compute it using predetermined concepts. $\delta' = f(\delta)^2$

This concept greatly limits the programmer's influence and can be detrimental or beneficial depending on the scenario. There are no states in a functional program, and thus there are no assignment variables. However, this allows us to create code that has the following benefits. Hughes

1. The concept of modularity: A great aspect of Functional language is the level of abstraction associated. As we strictly describe what needs to be done, and rely on the compiler to create the low-level logic instead, our final code ends up being a collage of reusable functions. These can be reused to expand the functionality and are modular enough that changes to one of these components should not break the entire code base. As most languages deal with immutable data structures and lack of side effects, it's relatively easier to diagnose a solve an issue in an isolated manner and lets us enforce separations of concerns effectively.
2. The concise nature of FP: The code generally is much more concise than its imperative counterpart. Most code consists of a description of functions, data structures, and an immutable declaration of variables. A large portion of often reused code or boilerplate that is associated with imperative programming is discarded to achieve the maximum resemblance to pure mathematical notions, that inspired functional programming's origin.

Functional programming excels when working closely with data and large magnitude of computation. It's safer approach and more concise nature allows programmers to create reliable software. In Trends in Functional Programming 13th International Symposium (**turner2004total<empty citation>**), is a great breakdown of how functional programming is used in various sections of the industry, revolutionizing and setting a standard in many cases.

Functional programming began with LISP language in the very essence, developed by John McCarthy in 1960 (**mccarthy1960lisp<empty citation>**), where McCarthy aimed to create a language that would work with symbolic data. What began as the basis, quickly expanded into multiple concepts that describe a modern functional programming language:

- The Lambda Calculus (**Rosser1941-ROSCAT-17**<**empty citation**>) depicted typeless theory and theorems that are rudimentary building blocks of FP, describing how the functions interact to create a programming logic. It was further explored as the formation of Algol 60 (**backus1963revised**<**empty citation**>).
- Development of lazy evaluation contributed to SASL and polymorphism attributed in development to ISWIM. Various evolution in the field happened in Edinburgh throughout the late 1960s and 1980s.
- General development of more popular languages such as Miranda and Haskell that brings the current state of FP to be lazily evaluated, typed with algebraic types with polymorphism, have functions that are high order and have features such as pattern matching inbuilt.

A Example appendix

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante.

Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

A.1 A subsection

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

A.1.1 A subsubsection

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.