Programming Languages

Reflection of languages

Zain Al – hassan - 17106353

2020

Contents

[Introduction: 2](#_Toc29841059)

[Description of programming languages: 2](#_Toc29841060)

[Smalltalk 2](#_Toc29841061)

[Python 2](#_Toc29841062)

[JavaScript 3](#_Toc29841063)

[Clojure 3](#_Toc29841064)

[Haskell 3](#_Toc29841065)

[Comparison of languages 3](#_Toc29841066)

[Readability and Writability: 3](#_Toc29841067)

[Reliability: 4](#_Toc29841068)

[Popularity: 4](#_Toc29841069)

[Evaluation and Reflection: 4](#_Toc29841070)

[References: 5](#_Toc29841071)

# Introduction:

The aim of this report is to compare and evaluate five different programming languages learnt. This report will not only assess the key features of each programming language but will also discuss the experience involved with utilizing each of them and which programming language is superior and how so. The programming languages used are as follows: Python, Smalltalk, JavaScript, Clojure and Haskell

# Description of programming languages:

Description of each of the programming language with an overview of the features of each programming language.

## Smalltalk

Smalltalk is a pure object-oriented programming language that is typed dynamically. It uses objects, which are instances of a classes which themselves are objects too. It is a multi-paradigm language, meaning it is suited equally for more than one paradigm. It supports object-oriented and symbolic programming. Moreover, there are no primitive types (int, Boolean, float etc.), meaning that variables are type-less. In Smalltalk methods are not called but objects send messages to other objects to look for methods in that object. (Medium, 2019) If the method is not found in that object, that message is redirected to the superclass. Additionally, Smalltalk supports inheritance, polymorphism and encapsulation. Encapsulation, prevents objects being modified directly, improving maintainability throughout the code. It allows automatic garbage collection; it deletes unused objects and frees up space. Smalltalk also allows the use of control structures such as if, else, loops etc. (MMU. (2019)).

## Python

Python is an object-oriented programming language that is dynamically typed. Its interpreter reads one line of code at a time, translating it into byte code (low level machine language) before execution. This also aids with code debugging (GeeksforGeeks. (2020)). Python also greatly emphasises on the readability of the code. Both basic variable types and containers such as lists, and arrays are objects. Python allows automatic garbage collection (efficient use of memory) and the use control structures (if, else, loops etc.). Python is a multi-paradigm programming language, supporting both object-oriented and imperative programming. Python includes primitive types such as integers, floats, Booleans and strings. It supports polymorphism, encapsulation and inheritance (MMU. (2019)).

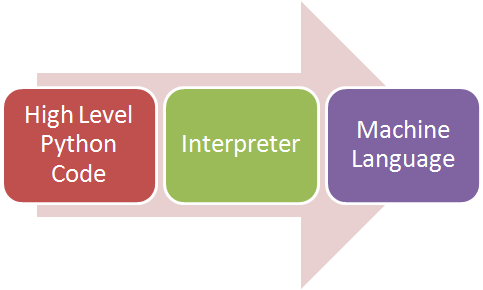


Figure 1: Image representing code conversion (Malik, 2020)

## JavaScript

JavaScript is an object-based programming language. However, JavaScript does use functions (which are also objects) (phpied.com, 2020). JavaScript is mainly used to make webpages more user interactive by adding animations, clickable buttons popup dialogs etc (MDN Web Docs, 2019). JavaScript allows inheritance. JavaScript is also a multi-paradigm programming language, supporting both object-oriented and functional programming. JavaScript includes a large number of primitive types which include Boolean, Number, string, symbol, undefined and null (Medium. (2020)). JavaScript allows the use of control structures. It supports automatic garbage collection (efficient use of memory). The language is weakly typed, replicating polymorphism which allows multiple implementations under one defined interface.

## Clojure

Clojure is a functional programming language which emphasises on being dynamic. Clojure uses JVM (Java Virtual Machine) to function and it consists of immutable persistent data structures, meaning they preserve the previous version even after being modified. Clojure is a symbiotic language which was designed for concurrency (two or more tasks are executed at the same time using multi-core) By taking advantage of JVM’s capabilities, Clojure is able to support a faster type of polymorphism. (Clojure.org. (2020)). Clojure consists of large number of data structure sets; nil, numbers, longs, ratio, contagion etc. Clojure also supports control structures (Practicalli.github.io, 2020).

## Haskell

Haskell is a purely functional programming language (no side effects involved) that is declarative, meaning it does not require a control flow to be described. It is completely immutable and if evaluated; expressions cannot be changed. Thus, meaning basic refactoring. It does not have variables and it doesn’t support control structures such as for and while loops. Haskell is statically typed (Haskell.org. (2020)). Haskell tools include ghc which is responsible for optimising the compiler which produces fast native code and GHCI (interpreter and debugger). It also includes automatic garbage collection (MMU. (2019)).

# Comparison of languages

## Readability and Writability:

Smalltalk’s expressive nature increases the readability, as its simple syntax makes coding easy to understand and quick to implement (Web.cecs.pdx.edu. (2020)). Similar to Smalltalk, Python is also popular for its readability, it uses more expressions than symbols and the syntax design is kept simpler, making it much easier for beginners to get used to the language and just like Smalltalk, the simplicity, syntax design and data types makes it easier to write code (Nedbatchelder.com, 2020). As compared to both of these languages, JavaScript is the least expressive language because of its syntax design and the fact that there are multiple ways of writing code which performs the same tasks, some of which can be difficult to read. This also affects the writability of the language as the complex coding style, syntax and reduced expressiveness makes it harder to code in JavaScript (Tsonev, 2015). Similar to JavaScript, Clojures’ readability does not compare to Smalltalk and Python. Visually, Clojure is very similar to JavaScript and the use nested operations in Clojure can make the code hard to understand (Vvvvalvalval.github.io, 2018). Haskell provides developers with tools that can help make the code more readable, self-evident and easier to write. However, unlike Python and Smalltalk, Haskell relies more on the symbols which affects the writability of the code (Williame.github.io, 2020).

## Reliability:

In Smalltalk and Python, type checking is fully absent, affecting the reliability but on the other hand, consistent style of writing (everything is an object), the expressiveness, simplicity and exception handling proves that Smalltalk and Python are fairly reliable programming languages. Also, code is executed by each line, making debugging code easier and this makes Python more reliable than Smalltalk. As compared to Python, JavaScript is a poorly designed language (EDUCBA, 2020). There are three characteristics that affects the reliability of JavaScript and Clojure in an undesirable way; simplicity, expressivity and syntax design. Overall, Haskell is quite reliable due to the fact that it is simple to use and its typing approach. It does not allow functions to have any side effects making it more reliable than Clojure. Haskell involves type checking at both compile and run time unlike Smalltalk and Python (Mta.ca, 2020).

## Popularity:

JavaScript is the most popular language because it provides interactivity to websites i.e. transitions, scroll, object movement etc. It is used in Front-end as well Back-end in web development. Python comes in par with JavaScript when it comes to popularity. Python is mainly popular for its productivity, simple coding syntax and reliability. Python requires less time and lines of code to perform a set of operations as compared to any other language. Haskell used to be quite popular in the past but the facilities it provides are different from what is needed by mainstream programmers hence why Haskell started losing popularity despite being a reliable language. Smalltalk used to be the most popular OOP language in the 1990, the introduction to other OOP languages such as Python caused its popularity to its decline. Clojure initially gained popularity because of the fact the actual language design was good and the adaptation of another data structure. However, the lack of emphasis on other aspects of the language was the reason behind its decline in popularity.

# Evaluation and Reflection:

Throughout all the languages that were studied in this module, I found JavaScript to be easiest language to work with due to the fact I have worked with the language previously in assignments, experience and knowledge of the expressions, syntax and how the language worked overall proved highly beneficial. From the languages I learnt during this module, I found Python and Smalltalk to be the easiest. As a beginner to the languages, I was quickly able to grasp how both programming languages worked due to its simplicity in expression names and syntax design. I struggled with both, Clojure and Haskell due to the fact that I have never worked with a functional language before and compared to Smalltalk and Python, both of these programming languages were harder to understand. I found Clojure to be the most challenging language because Haskell was somewhat self-evident as compared to Clojure. After working with all five of these languages I found JavaScript and Python to be my strongest two languages out of the rest which are both Object Based and Object-Oriented Programming respectively. This was due to the similarities in concepts and design patterns that I have learnt through different languages.

Working with all five of these languages has greatly enhanced my knowledge, understanding and experience as programmer which will be of grave importance in the future especially while looking for a job in the future. This has given me a better understanding of where my skills as a programmer are.

# References:

Clojure - Runtime Polymorphism. [online] Available at: https://clojure.org/about/runtime\_polymorphism [Accessed 13 Jan. 2020].

Haskell Language. [online] Available at: https://www.haskell.org/ [Accessed 13 Jan. 2020].

JavaScript Essentials: Types & Data Structures. [online] Available at: https://codeburst.io/javascript-essentials-types-data-structures-3ac039f9877b [Accessed 13 Jan. 2020].

Malik, F. (2020). *Everything About Python — Beginner To Advanced*. [online] Medium. Available at: https://medium.com/fintechexplained/everything-about-python-from-beginner-to-advance-level-227d52ef32d2 [Accessed 12 Jan. 2020].

MDN Web Docs. (2019). Introduction. [online] Available at: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction [Accessed 13 Jan. 2020].

MMU. (2019) *Lecture 10 – Haskell* [Online] [Accessed on 11th January 2020]

MMU. (2019) Lecture 2 - Smalltalk/Pharo [Online] [Accessed on 9th January 2020]

MMU. (2019) *Lecture 3 – Python* [Online] [Accessed on 10th January 2020]

MMU. (2019) *Lecture 5 – JavaScript* [Online] [Accessed on 10th January 2020]

MMU. (2019) *Lecture 8 – Clojure* [Online] [Accessed on 10th January 2020]

Mta.ca. (2020). Functional Programming with Haskell. [online] Available at: https://www.mta.ca/~rrosebru/oldcourse/371199/haskell/paper.htm#relia [Accessed 13 Jan. 2020].

Nedbatchelder.com. (2020). Ned Batchelder: Python’s misleading readability. [online] Available at: https://nedbatchelder.com/blog/201801/pythons\_misleading\_readability.html [Accessed 13 Jan. 2020].

phpied.com. (2020). 3 ways to define a JavaScript class. [online] Available at: https://www.phpied.com/3-ways-to-define-a-javascript-class/ [Accessed 13 Jan. 2020].

Practicalli.github.io. (2020). Control flow · Clojure, Practicalli. [online] Available at: https://practicalli.github.io/clojure/basic-clojure/control-flow.html [Accessed 13 Jan. 2020].

Python Features - GeeksforGeeks. [online] Available at: https://www.geeksforgeeks.org/python-features/ [Accessed 13 Jan. 2020].

Smalltalk is not a perfect language. What are its shortcomings?. [online] Available at: https://itnext.io/smalltalk-is-not-a-perfect-language-what-are-its-shortcomings-72ac93b1ef0d [Accessed 13 Jan. 2020].

Smalltalk Overview. [online] Available at: http://web.cecs.pdx.edu/~harry/musings/SmalltalkOverview.html [Accessed 13 Jan. 2020].

Tsonev, K. (2020). *Rethinking JavaScript readability*. [online] krasimirtsonev.com. Available at: https://krasimirtsonev.com/blog/article/rethinking-javaScript-readability [Accessed 13 Jan. 2020].

Vvvvalvalval.github.io. (2018). *Val on Programming: So yeah, about Clojure's syntax...*. [online] Available at: https://vvvvalvalval.github.io/posts/2018-01-06-so-yeah-about-clojures-syntax.html [Accessed 13 Jan. 2020].

Williame.github.io. (2020). Programming Language Readability : Will Edward's Programming. [online] Available at: https://williame.github.io/post/18319031919.html [Accessed 13 Jan. 2020].