

CSE 341 Programming
Languages HW#1

Ömer Faruk Akduman
1801042094

Index

Introduction	2
The process of choosing programming languages that will be compared.....	2
Considered Programming Paradigm.....	2
What are the “Programming Paradigm”	2
Considered Popularity	4
History of compared languages	5
History of C Programming Language	5
History of Lisp Programming Language.....	5
Conclusion.....	5
Usage Areas.....	5
Usage area of C Programming Language.....	6
Usage area of Common Lisp	6
Some Important Companies & Projects	6
Some important Companies using Lisp.....	6
Some important Companies using C	7
Conclusion compared usage areas.....	7
Syntax.....	8
Basic Syntax Comparision	8
Print “Hello World” In C.....	8
Print “Hello World” In Common Lisp	8
Mathematical Operations.....	8
Data Types.....	9
Data Types of C.....	9
Immutable data.....	9
Exception Handling	10
Exception Handling in C	10
Exception Handling in Common Lisp.....	10
Efficiency	10
Conclusion.....	11

Introduction

This is an essay about analyzing two programming languages that I know. While writing this article, I first chose two programming languages considering certain criteria. Then I briefly summarized the history of these programming languages. Then I compared the usage areas because usage areas are an important criterion to better understand the character of a programming language.

Then I did a syntax comparison to make it more concrete, followed by data types. I also talked about Exception Handling and gave examples of it. After that I compared it in terms of efficiency and do some run-time tests on the same machine, which were quite interesting.

Since I think the learning curve is quite important for a language, I wrote down some my friends' thoughts about the learning curve of these programming languages. I compared their popularity up to date, and finally I gave examples of important, notable projects using this program's languages.

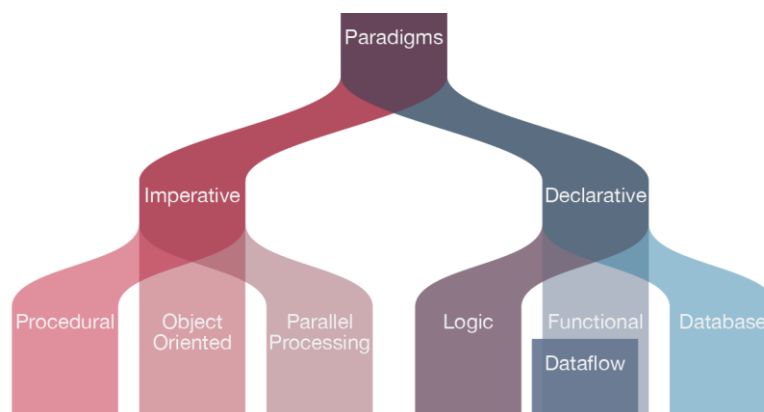
In other words, I tried to examine these two programming languages in a multidimensional way by using our textbook "Concepts of Programming Languages ELEVENTH edition by Robert W. Sebesta" and the sources I mentioned in the bibliography.

The process of choosing programming languages that will be compared

Considered Programming Paradigm

What are the “Programming Paradigm”

Programming paradigm is about “how you design or organize your algorithm - problem solving style-”. It is not about any syntax feature or something. It is just how you structure your code and how you reason about your program you are writing. So how your code is organized that we can called programming paradigm. There are many programming paradigms that you can gather under 2 headings.



Source: [https:// www.watelectronics.com/](https://www.watelectronics.com/)

Since programming paradigms are very important for programming languages, I preferred that the 2 programming languages that I will compare have different programming paradigms like one of them is Imperative “Procedural” the other one is Declarative “Functional”. Now I can take a more detailed look at the programming paradigms I have chosen.

Procedural:

If you want to write procedural style code, you should write a bunch of execution steps and tasks. Also, procedural programming paradigm has top to bottom code execution. In procedural programming you can write code more modular and organized than imperative.

Examples of Procedural languages:

FORTRAN, COBOL, ALGOL, BASIC, C and Pascal.

Functional:

If you want to write functional programming style code, when you write your code you should put functions at the center of programming. Programming languages that have functional programming paradigms are generally designed for symbolic computation and list processing applications. Also, against procedural programming paradigm generally don't fully support flow controls, functional programming paradigm directly uses functions and function calls. Functional programming languages are based on the concept of mathematical functions that are using recursion processing and conditional expressions to execute code.

In Object Oriented Programming paradigm there are some popular concepts such as abstraction, encapsulation, inheritance, and polymorphism, in functional programming paradigm these species also support like OOP paradigm.

Another important qualification functional programming paradigm is it uses immutable data. Immutable data means if you have data that you cannot change or cannot modify these data are immutable like String in Java. Another hand if you can modify or change data that is called mutable data. At the first glance Immutable data can be look bad, but it has also some advantages like speed of access data and thread-safe “threads that are using same data, if there is shared data structure, immutable data can be more safe than mutable data”.

The main idea of functional programming is to simulate the mathematical functions

The other important concept about functional programming paradigm is pure functions, pure functions mean there is no hidden parameter, all parameters are visible, and output also should declared as output.

An example of pure function:

```
Pure Func(a, b, c, d) {  
    Return a + b + c + d;  
}
```

An example of impure function:

```
Int a, b, c;  
...
```

```

Int d;

Pure Func() {

    d = a + b + c;

}

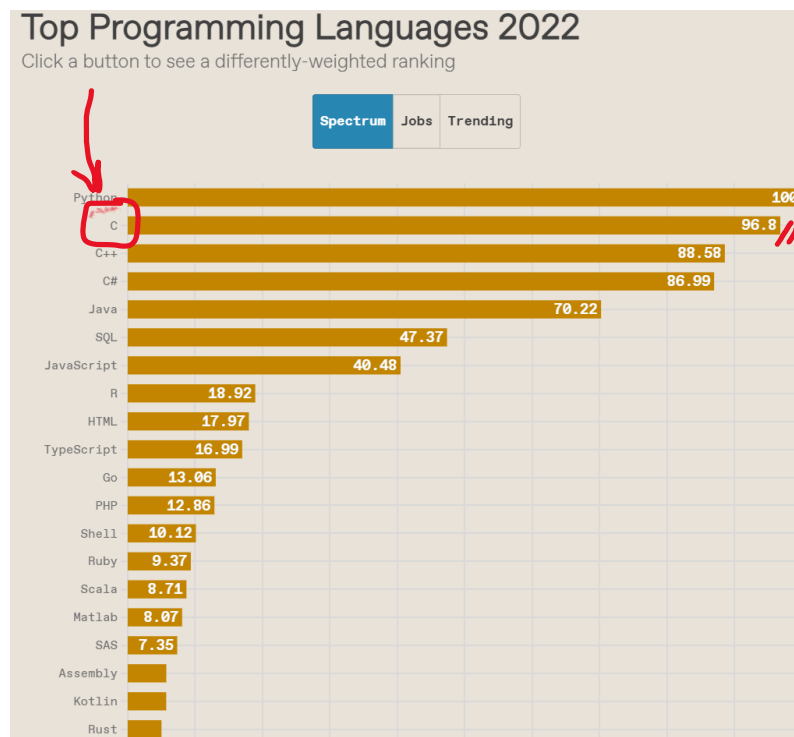
```

Here, are some most prominent Functional programming languages:

Haskell, SML, Clojure, Scala, Erlang, Clean, F#, ML/ **Lisp** / Scheme, XSLT, SQL, Mathematica

Considered Popularity

Popularity is another important criterion in selecting programming languages because generally if a programming language is popular, it can be also more useful in nowadays conditions.



Source: IEEE

Nowadays **C** is very popular programming language, according to IEEE (Institute of Electrical and Electronics Engineers) **C** is very popular. So that I selected first **C** in procedural paradigm languages.

In the other hand, in functional programming languages, **Lisp** is one of the most popular functional programming languages.

28	Rank	Ada
29		Lua
30		Lisp
31		PL/SQL
32		Dart

Source: TIOBE

Finally, my two programming languages that are I will analyze are **C** and **Lisp**. C represents procedural programming paradigm and Lisp represents functional programming paradigm.

History of compared languages

History of C Programming Language

In 1970s C programming language is developed by American computer scientist Dennis M. Ritchie at Bell Laboratories (formerly AT&T Bell Laboratories). C was designed to write operating systems for minicomputers, such as the DEC PDP 7, which had very limited specialties compared with the mainframe computer. After C programming language developed, C would also form the basis of Unix. *Source: Britannica*

History of Lisp Programming Language

LISP, in full list processing, a computer programming language developed about 1960 by John McCarthy at the Massachusetts Institute of Technology (MIT). LISP was founded on the mathematical theory of recursive functions (in which a function appears in its own definition). *Source: Britannica*

Conclusion

C developed for write operating systems, but Lisp created for data processing and Lisp was developed 10 years before the C language.

Usage Areas

Today, there are thousands of programming languages and almost all of them have been developed to solve a problem, that is, each one has a usage area, they can be used in some mobile industry, some web technologies, some operating systems and many more. Others can be used in more than one area.

Usage area of C Programming Language

C is one of the most popular languages today. It is relatively easy to learn and apply. Many features such as speed, reliability and dominance of the market make C programming language one of the most widely used languages.

Writing programming language: *“CPython is the reference implementation of the Python programming language. Written in C and Python, CPython is the default and most widely used implementation of the Python language.”* source: Wikipedia

Writing Operating System: Unix is the first operating system that designed in high level programming languages that C programming languages. After designed Unix, Windows and various Android applications and operating systems designed in C.

In Embedded Systems: Thanks to C programming language is closely related to machine hardware, it is the optimum choice for writing Embedded Systems.

GUI “Graphical User Interface”: In gui systems C programming language is not so popular but if a developer wants to develop a gui system then developer can use C. For example, gui is very important in the photoshop industry, Adobe Photoshop is one of the leading companies in the industry and Adobe Photoshop’s gui is developed with the C programming language.

Usage area of Common Lisp

Lisp is after Fortran “FORMula TRANslating” second oldest programming language still use. Lisp is still using in academia projects, both simple programming projects and highly complicated projects like AI to machine learning and quantum computing. Beyond the university walls Lisp is using by the following: symbolic AI programmers, embedded systems programmers; small or understaffed programming teams. Even you can use Common Lisp in web-based applications.

Some Important Companies & Projects

Some important Companies using Lisp

- Boeing uses the Allegro NFS Server in the onboard network system of Boeing 747 and 777 aircraft (source: [video, ELS 2015](#)).
- [Grammarly](#), create an English language writing-enhancement platform. [Running Lisp in Production](#)

Full list is: <https://common-lisp.net/lisp-companies>

Some important Companies using C

List of C Companies: Overview

Companies That Use C	Who Uses C at This Company?	What Does This Company Use C For?	Estimated Number of Employees
Apple	Software Data Engineer, System Architect	Building visualization tools	More than <u>147,000</u>
Facebook	Software Engineer Manager, Hardware Platforms Architect, System Validation Engineer	Scripting and automation	More than <u>58,604</u>
Google	Software Engineer, Hardware Engineer, Engineer Manager	Write product or system development code	More than <u>135,301</u>
Mastercard	Senior Software Engineer, Lead Software Development Engineer, Lead Database Engineer	Designing databases	More than <u>21,000</u>
Microsoft	Software Engineering Lead, Senior Electrical Engineer, Senior Coherence Verification Engineer	Develop software	More than <u>181,000</u>
Nvidia	Software Engineer, Senior Systems Software Engineer	Develop product code	More than <u>18,975</u>

Source: <https://careerkarma.com/blog/who-uses-c/>

Conclusion compared usage areas

When we compare the companies, projects and usage density using these programming languages, we can clearly see that the C programming language is superior in popularity to Common Lisp today. But this does not mean that C is superior to Lisp.

Syntax

All languages must have a syntax. Both spoken languages and programming languages. For example, there are 616,767 words in Turkish, if there are no rules, how can an understanding be achieved between people who speak Turkish? Without these rules, communication and agreement cannot be achieved. These rules are also called syntax.

Basic Syntax Comparison

Print “Hello World” In C

```
#include <stdio.h>    *1//header of the library
*2
*3int main() *1/*function: int main()*/ {
*4printf("Hello World!");
return *50;
}
```

*2 Blank line: C ignores white space. But we use it to make the code more readable.

*1 Comment lines

*4 printf(): is a function used to print text to the terminal. In here prints “Hello World” to terminal.

*3 Returning type

*5 Returning value

Print “Hello World” In Common Lisp

```
*1; gnu clisp 2.49.60
*3(*2print "Hello World!") *3
```

*3 Parentheses is very important in lisp

*1 Comment line, in here specifies the version

*4 print is a function used to print text to the terminal. In here prints “Hello World” to terminal. Actually print object & optional output-stream => object

Mathematical Operations

Type

Common Lisp

C Programming Languages

Addition	(+ 2 3)	2 + 3
Multiply	(* 2 3)	2 * 3
Division	(/ 10 5)	10 / 5
Subtraction	(- 10 5)	10 - 5
Mod	(% 10 5)	10 % 5
	(Prefix notation)	Infix notation

Data Types

Data Types of C

Data Type	Example of Data Type
Basic Data Type “Primary”	Floating-point, integer, double, character.
Derived Data Type	Union, structure, array, etc.
Enumerated Data Type	Enums
Void Data Type	Empty Value
Bool Type	True or False

Source: Bbyjus.com

Basic Data Types: They are primitive data types further classified into: (a) integer types and (b) floating-point types.

Derived Data Types: They are derived from the basic data types after union some basic data types then they are called as one data type.

Enumerated Data Type: They are also again arithmetic type and assigned certain number.

Void Data Type: The type of specifier void indicates that no value is available.

Bool Type: Represent only true or false, false equals to int 0, except 0 other integers equal to true

Immutable data

In Imperative languages like c or c++ we use generally const (in java final etc.) specifier to build data like immutable data.

For example:

```
const a = 10;
```

In Common Lisp, all the data (such as primitive variable references, lists, arrays and hash tables) are mutable by default but we can give them read and write, read-only, write-only, or no access.

For example:

```
(defstruct student
  (name nil :type string :read-only t)
  (age nil :type (integer 0 100)))
s
```

Exception Handling

Exception Handling in C

The C programming language does not support exception handling nor error handling.

Exception Handling in Common Lisp

Common Lisp's one of the most features is its condition system. It's similar to exception handling mechanism in Java or Python but it's more comprehensive.

The condition handling system in LISP has three parts

- Signaling a condition
- Handling the condition
- Restart the process

Handling a Condition Process: You should firstly define the condition then you should write handlers lastly there is a restarting phase.

Efficiency

This part referenced *How to make Lisp go faster than C* paper by Didier Verna. This paper's aim is "Contrary to popular belief, Lisp code can be very efficient today: it can run as fast as equivalent C code or even faster in some cases." With some optimization.

C code:

```
void add (image *to, image *from , float val ){
    in t i ;
    const int n = ima -> n ;
    fo r ( i = 0 ; i < n ; ++i )
        to->data [ i ] = from->data [ i ] + val ;
}
```

Lisp code:

```
(defun add (to from val)

  (declare (type (simple-array single-float (*))

                to from))

  (declare (type single-float val))

  (let ((size (array-dimension to 0))) (dotimes (i size) (setf (aref to i) (+ (aref from i) val))))))
```

Optimization code:

```
(declaim (optimize (speed 0)

                   (compilation-speed 0)

                   (safety 3)

                   (debug 3)))
```

Results are:

Algorithm	Integer Image		Float Image	
	C	LISP	C	LISP
Assignment	0.29	0.29	0.29	0.29
Addition	0.48	0.48	0.47	0.46
Multiplication	0.48	0.48	0.46	0.45
Division	0.58	1.80	1.93	1.72

Figure 2: Execution Times (s), All Implementations

Efficiency conclusion: Appropriate data structures + type declarations and optimization settings
Common Lisp can be as much stronger as C programming languages.

Conclusion

As a result, C and Common Lisp programming paradigm differ from each other in many ways such as their functionality and the reasons for their design, and we see that even if we look at a programming language in many ways, we cannot say that one is superior to the other, but in today's conditions, we can say that C is a much more popular language than common lisp.

References:

How to make Lisp go faster than C paper by Didier Verna
http://www.iaeng.org/IJCS/issues_v32/issue_4/IJCS_32_4_19.pdf

www.watelectronics.com/

<https://common-lisp.net/lisp-companies>

<https://careerkarma.com/blog/who-uses-c/>