Art
01

CST8152
Compilers

Algonquin College

Computer Engineering Technology

CST8152 Compilers

Spring, 2023

Art 01

Prof. Paulo Sousa

Algonquin College

Computer Engineering Technology

CST8152 Compilers

Spring, 2023

Art 01

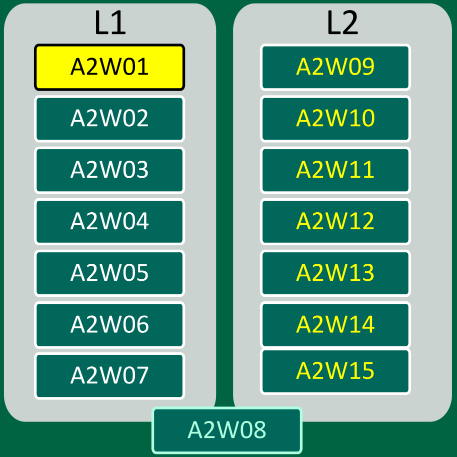
Introduction

Prof. Paulo Sousa

CST8152 — Compilers

Art 1: Introduction

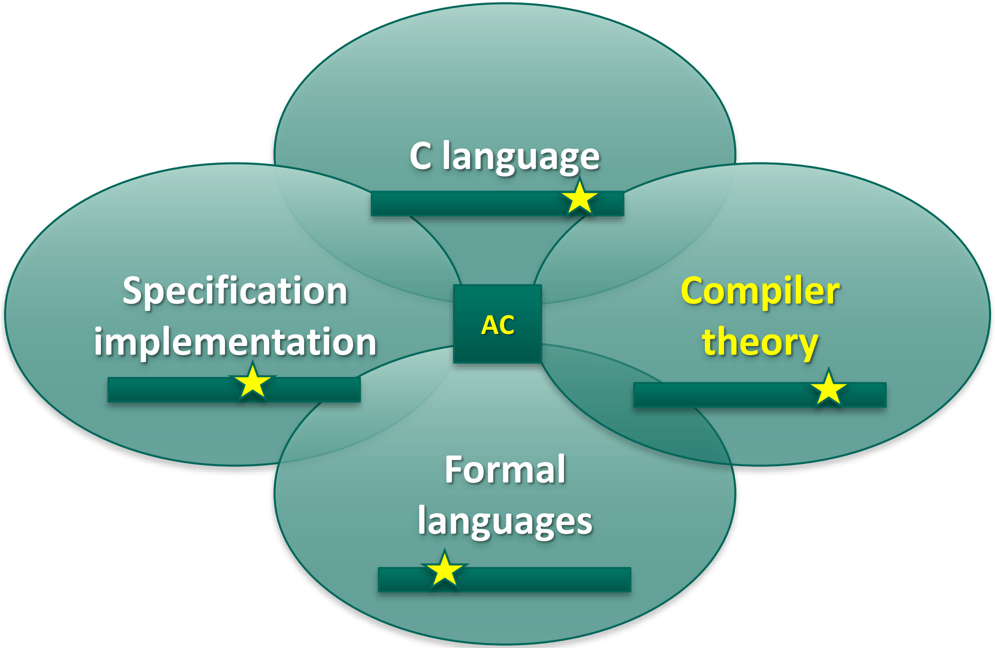
- *What is a Compiler?*
- *What about Programming Languages?*
- *Past, Present and Future.*



Compilers – Art 1

What is a Compiler?

Lets start…


C language
Specification implementation
AC
Compiler theory
Formal languages

1.1. Initial Concepts

- A **Compiler** is a program that runs on some computer architecture under some operating system and transforms (translates) an input program (source program) written in some programming language into an output program (target program) expressed in different programming language.
- A **Programming Language** is a notational system for describing computations in machine-readable and human-readable form.

Source: https://towardsdatascience.com/top-10-in-demand-programming-languages-to-learn-in-2020

1.1. Initial Concepts

- **Computation** in general is any process that can be carried by a computer. Programming Languages must provide two types of abstractions:
  - **Data abstractions** and
  - **Control abstractions**.

Note:

- Niklaus Wirth (Pascal language creator) defined a Programming language as: **Data structures + Algorithms.**
- Decades after, **OO** defined entities as composed by **properties** and **methods**.

Source: https://towardsdatascience.com/top-10-in-demand-programming-languages-to-learn-in-2020

1.2. Importance of Compilers

- **Compilers are used by all programmers**. "*A good craftsman should know his tools.*"
- **Compilers elements and techniques are used in almost every application**. See some domain-specific language (DSL).

Note:

- Writing a parser for XML, HTML, or some other structured data file is a common task.
- Scanning and parsing a command or user input line is a very common task.
- Looking for a specific word or sentence in a text is a very common task.

Source: https://towardsdatascience.com/top-10-in-demand-programming-languages-to-learn-in-2020

1.2. Importance of Compilers

- **Compilers are an excellent “capstone” or “focal” programming project.**
  - Writing a compiler requires an understanding of almost all of the basic computer science subfields.
- **Computer Science** specific topic.
  - It can demonstrate your proficiency in Computer Science.

Source: https://towardsdatascience.com/top-10-in-demand-programming-languages-to-learn-in-2020

1.3. Brief History

- **1840** – Analytical Engine (Charles Babbage).
- **1950** - FORTRAN, COBOL, Algol60, LISP.
- **1960** – PL/1, SNOBOL, Simula, BASIC
- **1970** – Pascal (1971), C (1972)
- **1980** - Ada, Modula, Smalltalk-80, C++, Objective C, Object Pascal, Eiffel, Oberon, Scheme, Logo
- **1990** – Java, Haskell, Javascript, PHP, Perl, Python, Ruby, Lua…
- **2000** - C#, Scala , F#, Groovy, Go, D, R, Clojure, Swift, Kotlin …
- **2010** – Rust (2015), Julia, …
- **2020** – GPT-3 (OpenAI), Platypus
- **2021-22** – OPT, …

Source: Wanner Bros.

1.3. Brief History

- **To think about the future:**
  - Facebook AI Creates Its Own Language In Creepy Preview Of Our Potential Future:
    - https://www.forbes.com/sites/tonybradley/2017/07/31/facebook-ai-creates-its-own-language-in-creepy-preview-of-our-potential-future/
  - The truth behind Facebook AI inventing a new language:
    - https://towardsdatascience.com/the-truth-behind-facebook-ai-inventing-a-new-language-37c5d680e5a7
  - OpenAI API:
    - https://openai.com/blog/openai-api/

Source: https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcRo3UP44TbbvFrlrRn43YRi3KwwK2yox3vmig&usqp=CAU

1.3. Brief History

- **To think about the future:**
  - GPT-3 Demo:
    - https://www.youtube.com/watch?v=8psgEDhT1MM
  - GPT-3 Paper:
    - https://arxiv.org/pdf/2005.14165.pdf
  - Kevin Lacker tests:
    - https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

Source: https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcRo3UP44TbbvFrlrRn43YRi3KwwK2yox3vmig&usqp=CAU

1.4. Recently

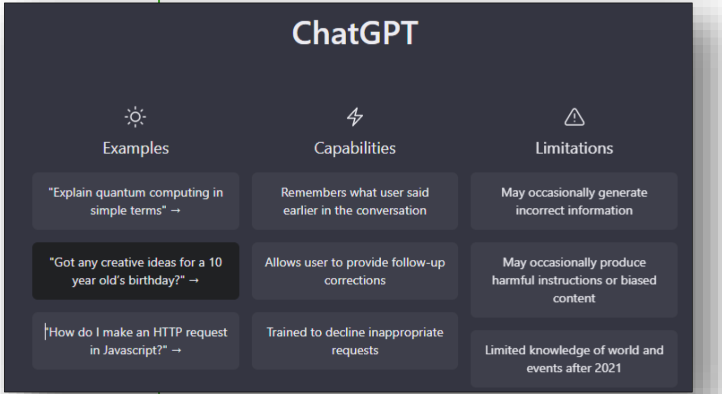
- IBM has released the “**CodeNet**” service.
  - https://research.ibm.com/blog/codenet-ai-for-code
- **OPT: Open Pre-trained Transformer Language Models:**
  - https://wandb.ai/telidavies/ml-news/reports/Meta-AI-Releases-OPT-175B-Set-Of-Free-To-Use-Pretrained-Language-Models--VmlldzoxOTQwOTU1
  - https://arxiv.org/abs/2205.01068

Source: https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcRo3UP44TbbvFrlrRn43YRi3KwwK2yox3vmig&usqp=CAU

1.4. Our “star”: ChatGPT

- **ChatGPT** was fine-tuned on top of GPT-3.5 using **supervised learning** as well as **reinforcement learning**.

Both approaches used human trainers to improve the model's performance. It uses the Proximal Policy Optimization (PPO) algorithms present a cost-effective benefit to trust region policy optimization algorithms and the models were trained in collaboration with Microsoft on their Azure supercomputing infrastructure.



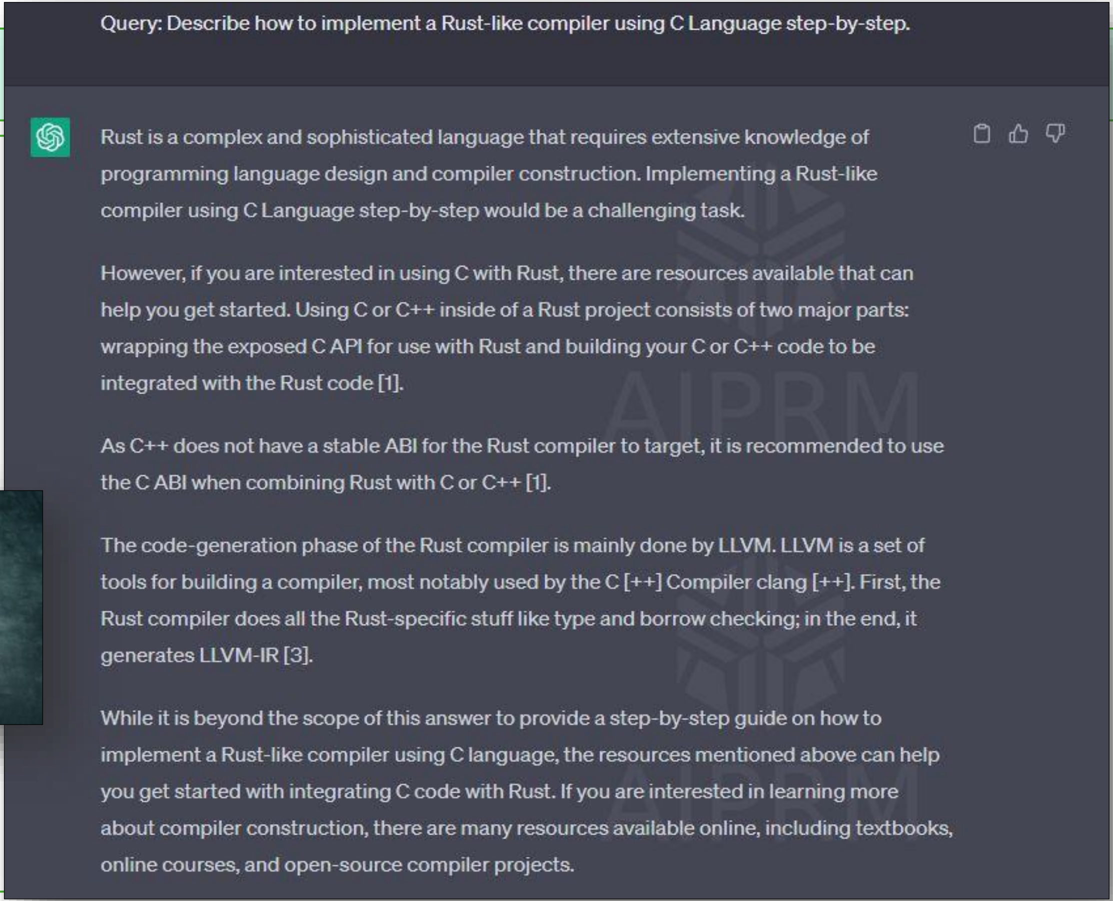
ChatGPT

Examples	Capabilities	Limitations
"Explain quantum computing in simple terms" →	Remembers what user said earlier in the conversation	May occasionally generate incorrect information
"Got any creative ideas for a 10 year old's birthday?" →	Allows user to provide follow-up corrections	May occasionally produce harmful instructions or biased content
"How do I make an HTTP request in Javascript?" →	Trained to decline inappropriate requests	Limited knowledge of world and events after 2021

➢ Basic Chat (Jan 6th 2023): https://openai.com/blog/chatgpt/

1.4. But…

- **Trying…**



Query: Describe how to implement a Rust-like compiler using C Language step-by-step.

Rust is a complex and sophisticated language that requires extensive knowledge of programming language design and compiler construction. Implementing a Rust-like compiler using C Language step-by-step would be a challenging task.

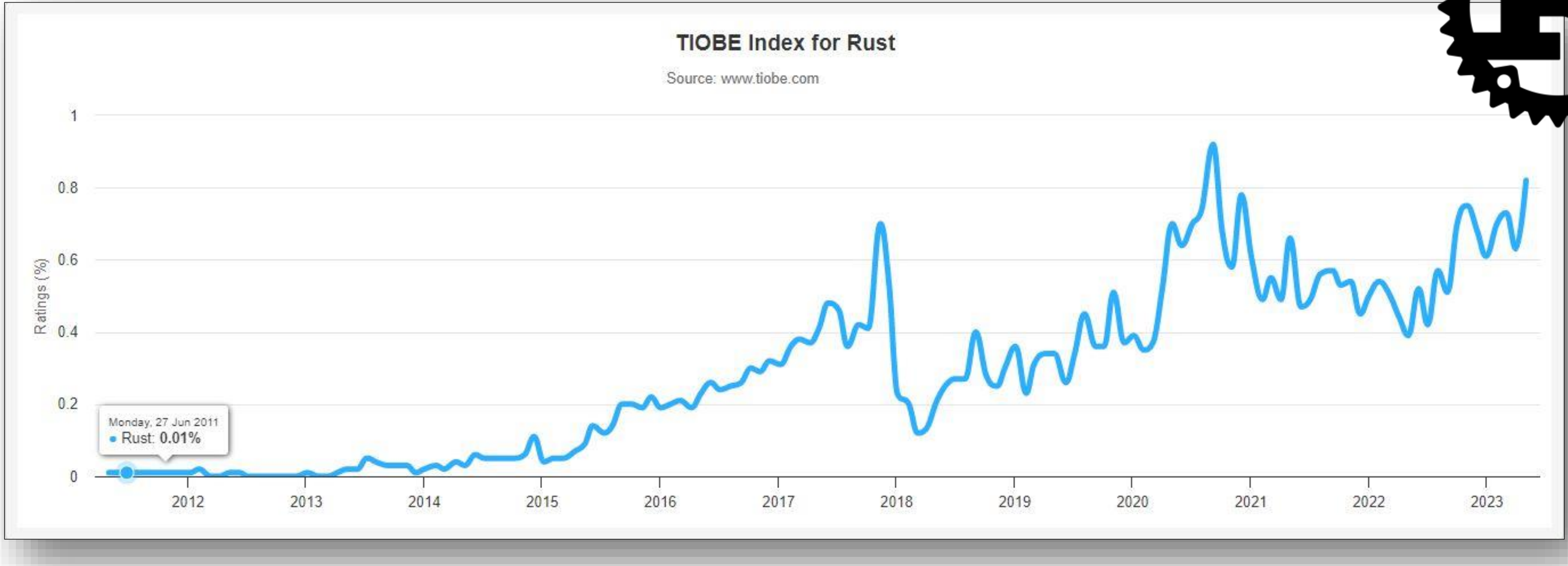
However, if you are interested in using C with Rust, there are resources available that can help you get started. Using C or C++ inside of a Rust project consists of two major parts: wrapping the exposed C API for use with Rust and building your C or C++ code to be integrated with the Rust code [1].

As C++ does not have a stable ABI for the Rust compiler to target, it is recommended to use the C ABI when combining Rust with C or C++ [1].

The code-generation phase of the Rust compiler is mainly done by LLVM. LLVM is a set of tools for building a compiler, most notably used by the C [++] Compiler clang [++]. First, the Rust compiler does all the Rust-specific stuff like type and borrow checking; in the end, it generates LLVM-IR [3].

While it is beyond the scope of this answer to provide a step-by-step guide on how to implement a Rust-like compiler using C language, the resources mentioned above can help you get started with integrating C code with Rust. If you are interested in learning more about compiler construction, there are many resources available online, including textbooks, online courses, and open-source compiler projects.

1.4. Rust…

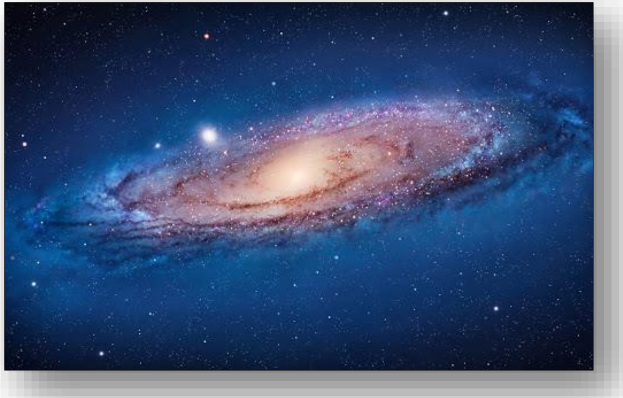

TIOBE Index for Rust
Source: www.tiobe.com
Ratings (%)
1
0.8
0.6
0.4
0.2
0
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
Monday, 27 Jun 2011
Rust: 0.01%

Compilers – Art 1

So... Let’s start...

1.4. Computational Paradigms

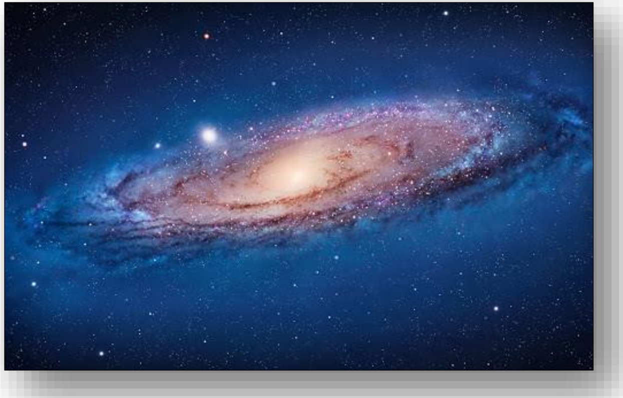
- **Imperative or Procedural Programming:**
  - FORTRAN, COBOL, ALGOL, BASIC, LOGO, PL, C, …
- **Functional Programming:**
  - LISP, Scheme, Miranda, Haskell, F#, Clojure.
- **Logic Programming:**
  - Prolog
- **Object-Oriented Programming:**
  - SIMULA, Smaltalk, C++, Objective C, Eiffel, JAVA, C#.
- **Scripting Languages:**
  - Perl, Python, Tcl/Tk, Javascript, Rexx, Visual Basic, PHP



Source: NASA

1.4. Special Purpose Languages

- **Database Query Languages:**
  - SQL
- **Simulation Languages:**
  - Simula, GPSS, SIMSCRIPT
- **Silicon Design Languages:**
  - VRML, VHDL,SystemC (C++), SpecC(C)
- **Graphics Design Languages**
  - GRAF
- **Real-time Languages:**
  - RT-FORTRAN, BCL, Embedded-C, Embedded Java



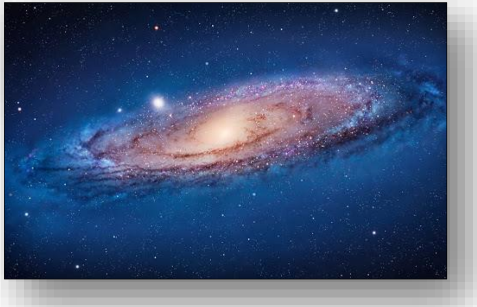
Source: NASA

1.4. DSL (Domain Specific Languages)

Definition:

A **domain-specific language** (DSL) is a computer language specialized to a particular application domain.

- **Note:**
  - This is in contrast to a general-purpose language (GPL), which is broadly applicable across domains.
- **Design Goals:**
  - Domain-specific languages are less comprehensive.
  - Domain-specific languages are much more expressive in their domain.
  - Domain-specific languages should exhibit minimal redundancy.

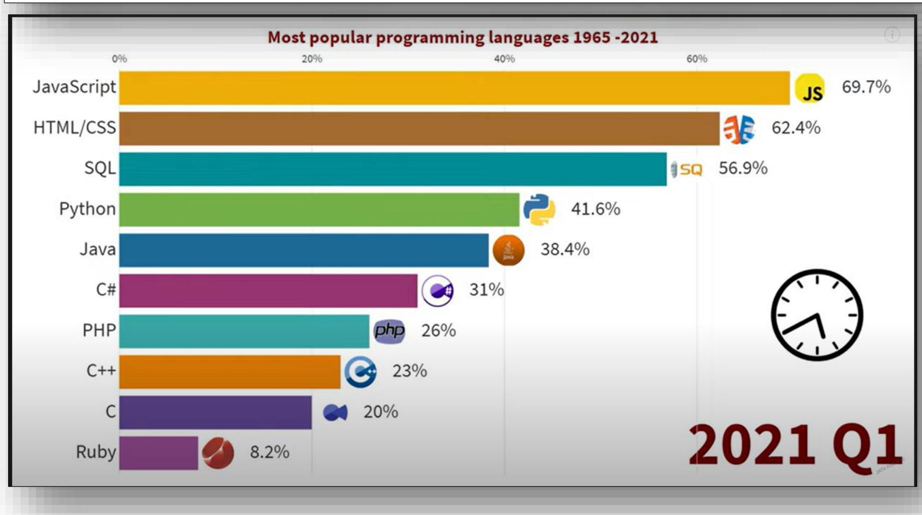


Examples:

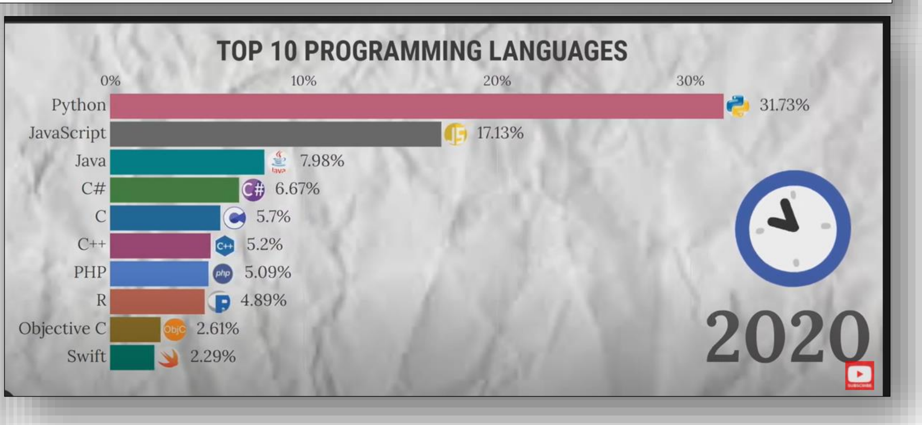
OS Shells, Wiki environments, OpenGL, Markup Languages…

Statistics – TIOBE (2022):

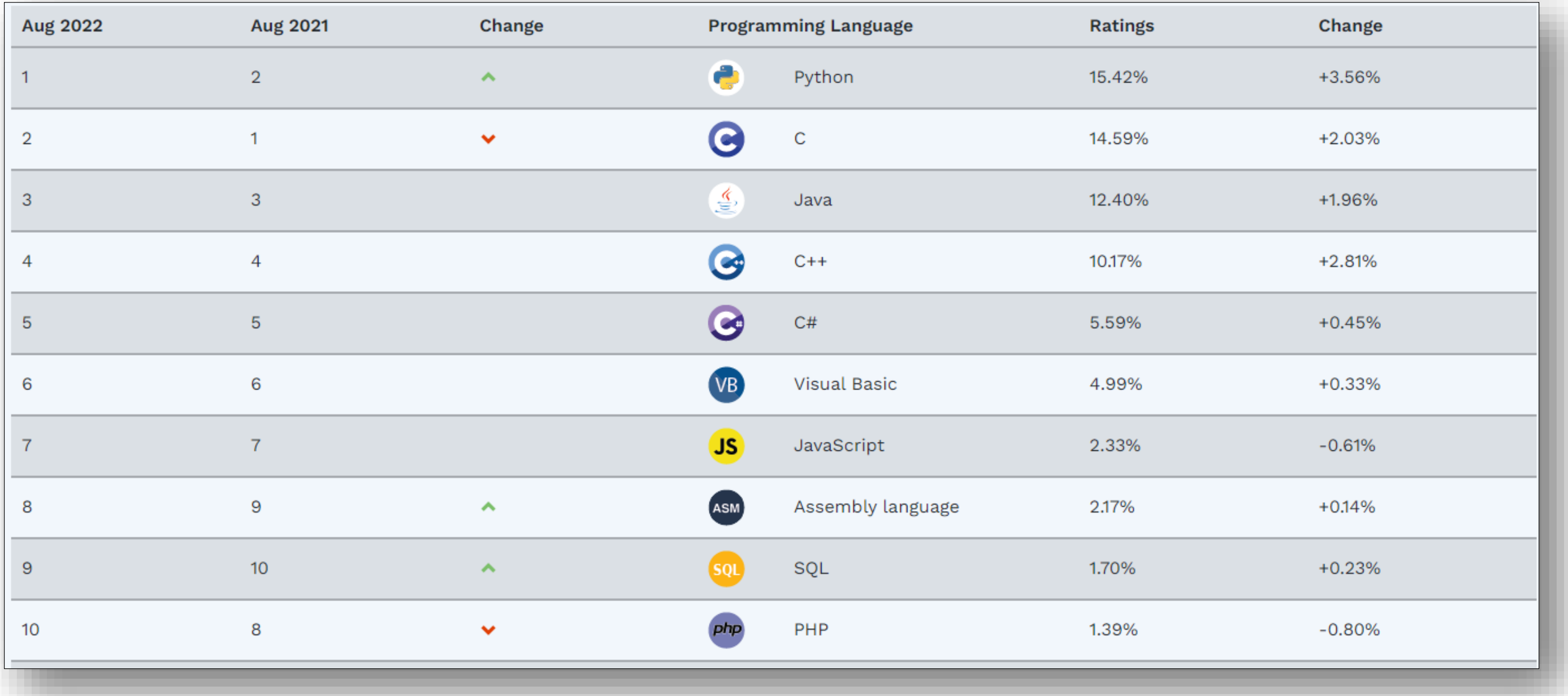
https://www.youtube.com/watch?v=qogEJSNZTPA



https://www.youtube.com/watch?v=2fZ4vHNvKj4

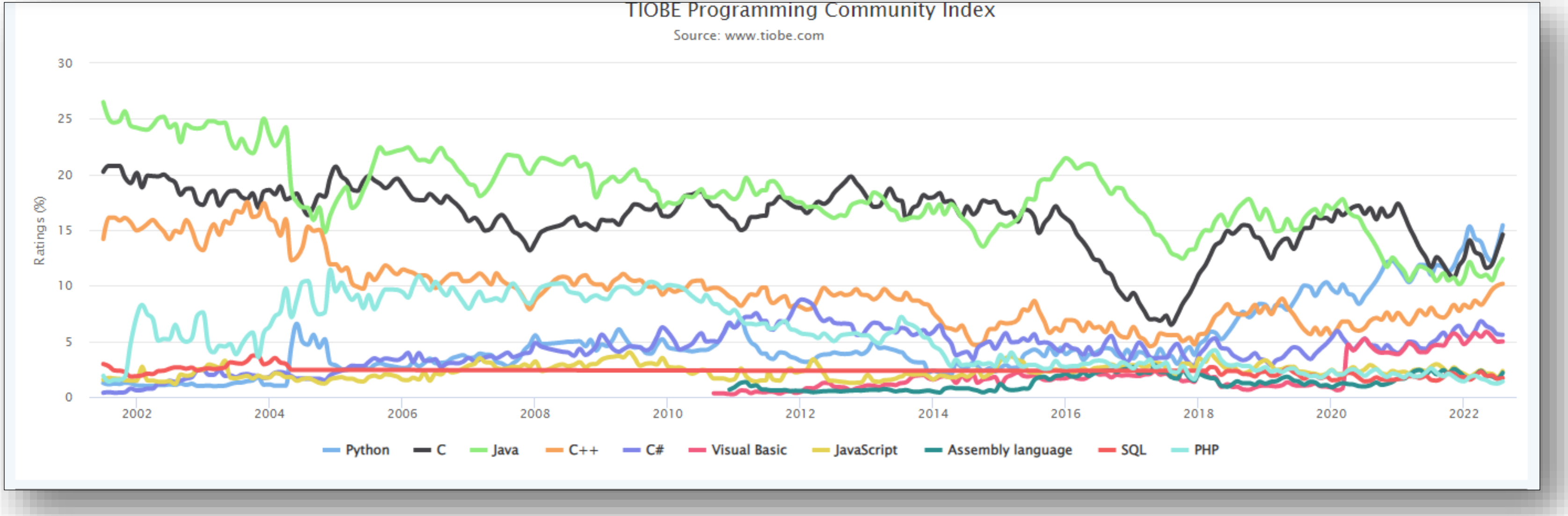


Statistics – TIOBE (2022-23):

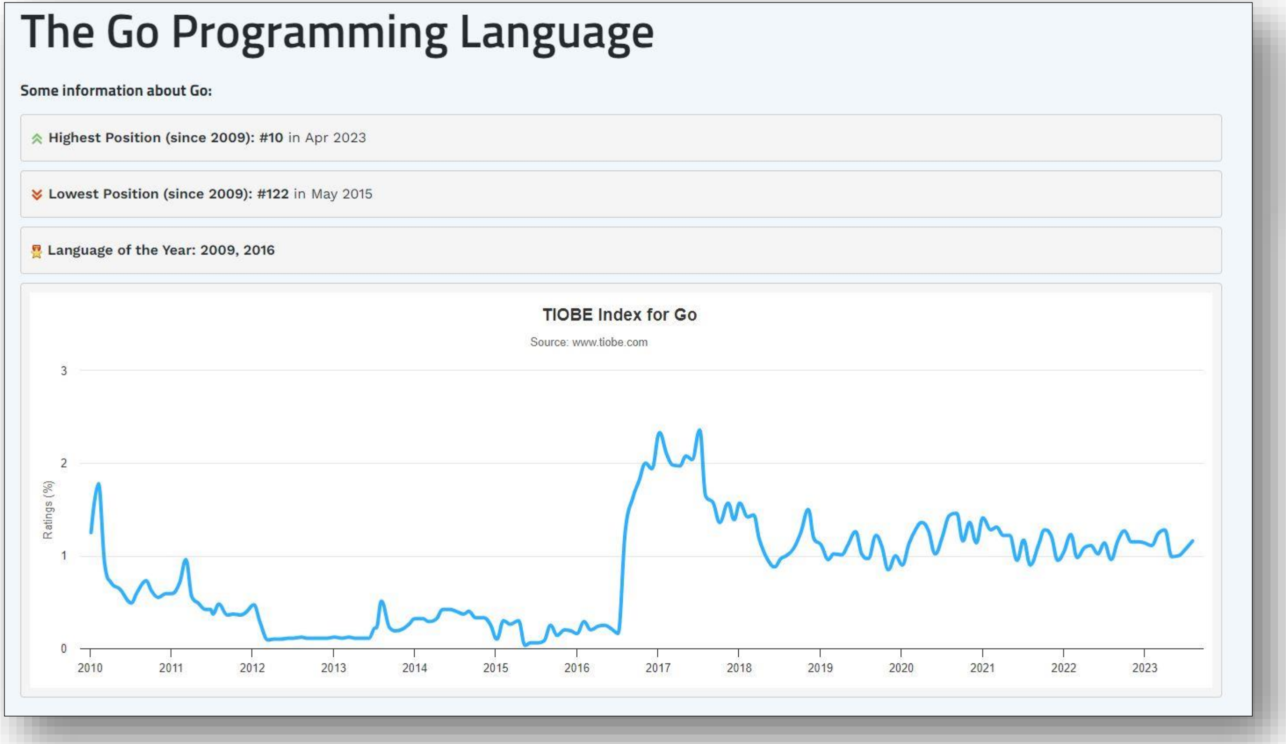


Aug 2022	Aug 2021	Change	Programming Language	Ratings	Change
1	2	▲	Python	15.42%	+3.56%
2	1	▼	C	14.59%	+2.03%
3	3		Java	12.40%	+1.96%
4	4		C++	10.17%	+2.81%
5	5		C#	5.59%	+0.45%
6	6		Visual Basic	4.99%	+0.33%
7	7		JavaScript	2.33%	-0.61%
8	9	▲	Assembly language	2.17%	+0.14%
9	10	▲	SQL	1.70%	+0.23%
10	8	▼	PHP	1.39%	-0.80%

Statistics – TIOBE (2023):



Statistics – Go (2023):



The Go Programming Language

Some information about Go:

Highest Position (since 2009): #10 in Apr 2023

Lowest Position (since 2009): #122 in May 2015

Language of the Year: 2009, 2016

TIOBE Index for Go

Source: www.tiobe.com

1.5. Some Definitions

- **Compiler Input:**
  - **Source program**
  - Configuration parameters or pragmatics (#pragma directives)
  - Source and Target Language Definitions
- **Compiler Output:**
  - **Target program**
  - Error messages
  - Information accompanying the target program – external symbol tables, cross- reference tables.

Target Program:

- High-Level Language
- Low-Level Code (Language)

Source: https://toCwardsdatascience.com/top-10-in-demand-programming-languages-to-learn-in-2020

1.5. Some Definitions

- **Target Low-Level Code Type:**
  - Pure Machine Code
  - Augmented Machine Code
  - Virtual Machine Code
- **Target Low-Level Code Format:**
  - Assembly or Pseudo-assembly Language Format,
  - Relocatable Binary Format
  - Memory-Image Format (Load & Go)

Source: https://toCwardsdatascience.com/top-10-in-demand-programming-languages-to-learn-in-2020

1.5. Some Definitions

- **Run-time Environment:**
  - Fully Static Environment
  - Fully Dynamic Environment
  - Mixed Environment – Stacked-based **environment**

Compiler Related Applications

- Editors, Word Processors, Command Interpreters, Formatting Printers, XML Parser, and almost all applications – big and small.

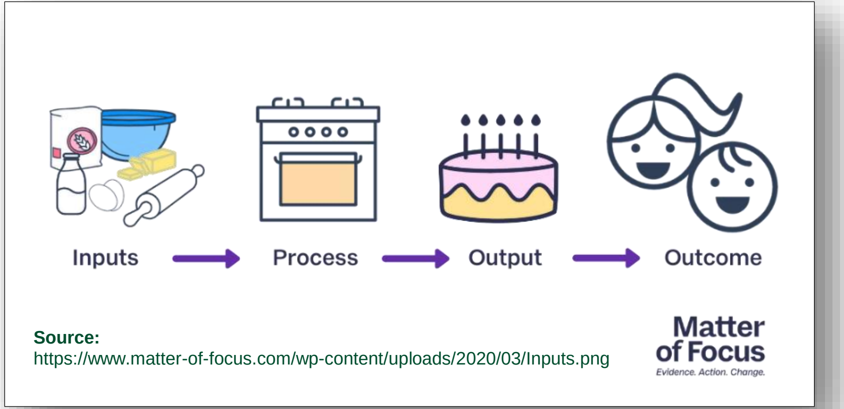
Source: https://toCwardsdatascience.com/top-10-in-demand-programming-languages-to-learn-in-2020

Compilers – Art 1

Concluding

Review

- *Compiler definition*
- *Basic overview (historical and trends)*
- *Computational paradigms*
- *Elements of compiler*



Source:
https://www.matter-of-focus.com/wp-content/uploads/2020/03/Inputs.png

Source:
https://static.wixstatic.com/media/7594af_51a81a8ccc5f418281f52c8bdd2dd618~mv2.jpg

Some Questions

1. *What is the importance of compilers?*
2. *Summarize the functionality of a compiler.*
3. *Identify some challenges to create a compiler.*

Source: https://static.wixstatic.com/media/7594af_51a81a8ccc5f41 8281f52c8bdd2dd618~mv2.jpg

Open questions…

- Any doubts / questions?
- How we are until now?

Image URL: https://cdn-az.allevents.in/events3/banners/26b150363d5757da8578fa6a1481585a368f12d4247adfe95837e6ea6c5ab2af-rimg-w1200-h549-gmir.jpg?v=1569691155

Compilers – Art 1

Thank you for your attention!

Contact: sousap@algonquincollege.com